Humboldt-Universität zu Berlin
Institut für Bibliotheks- und Informationswissenschaft

Berliner Handreichungen zur Bibliotheks- und Informationswissenschaft

Heft 438

Warenkorbanalyse für Empfehlungssysteme in wissenschaftlichen Bibliotheken

von
Veit Köppen

Warenkorbanalyse für Empfehlungssysteme in wissenschaftlichen Bibliotheken

von
Veit Köppen

Berliner Handreichungen zur
Bibliotheks- und Informationswissenschaft

Begründet von Peter Zahn
Herausgegeben von
Vivien Petras
Humboldt-Universität zu Berlin

Heft 438

Köppen, Veit

Warenkorbanalyse für Empfehlungssysteme in wissenschaftlichen Bibliotheken / von Veit Köppen. - Berlin : Institut für Bibliotheks- und Informationswissenschaft der Humboldt-Universität zu Berlin, 2019. – 112 S. : graph. Darst. - (Berliner Handreichungen zur Bibliotheks- und Informationswissenschaft ; 438)

ISSN 14 38-76 62

Abstract

Bibliotheken als Informationsdienstleister müssen im Datenzeitalter adäquate Wege nutzen. Mit der Durchdringung des Digitalen bei Nutzern werden Anforderungen an die Informationsbereitstellung gesetzt, die durch den täglichen Umgang mit konkurrierenden Angeboten vorgelebt werden. So werden heutzutage im kommerziellen Bereich nicht nur eine Vielzahl von Daten erhoben, sondern diese werden analysiert und die Ergebnisse entsprechend verwendet. Auch in Bibliotheken fallen eine Vielzahl von Daten an, die jedoch nicht genutzt werden. Schranken, wie der Datenschutz, werden häufig genannt, obwohl diese keine wirkliche Barriere für die Datennutzung darstellen. Die Analyse von anonymisierten Daten zur Ausleihe mittels Association-Rule-Mining ermöglicht Zusammenhänge in den Buchausleihen zu identifizieren. Die Ergebnisse können in den Recherche-Webangeboten den Nutzern zur Verfügung gestellt werden. So wird ein Empfehlungssystem basierend auf dem Nutzerverhalten bereitgestellt. Die technische Realisierung des Empfehlungssystems betrachtet die Datenerhebung, die Datenverarbeitung, insbesondere hinsichtlich der Data Privacy, die Datenanalyse und die Ergebnispräsentation. Neben der technischen Realisierung des Empfehlungssystems wird anhand einer in der Universitätsbibliothek der Otto-von-Guericke-Universität Magdeburg durchgeführten Fallstudie die Parametrisierung im Kontext der Data Privacy und für den Data Mining Algorithmus diskutiert. Damit liegt ein datengetriebenes Empfehlungssystem für die Ausleihe in Bibliotheken vor.

Diese Veröffentlichung geht zurück auf eine Master-Arbeit im postgradualen Fernstudiengang Bibliotheks- und Informationswissenschaft (Library and Information Science) an der Humboldt-Universität zu Berlin.

Eine Online-Version ist auf dem edoc Publikationsserver der Humboldt-Universität zu Berlin verfügbar.

Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Das Datenmanagement ist heute wichtiger denn je und insbesondere die Bibliothek als Informationsdienstleister, mit der Verpflichtung Menschen bei der Suche nach Information zu helfen, ist hierfür im Datenzeitalter essenziell [80, S. 6]. Bücher sind die wichtigste Ressource der Bibliotheken. Die Aufgabe des Bibliothekswesens ist wiederum die Nutzer[1] der Bibliothek mit den von ihnen gesuchten Informationen zusammenbringen [49]. Aufgrund der stetig anwachsenden Menge an Publikationen ist es daher folgerichtig, auch Unterstützungsmethoden aus dem Datenmanagement für die Bibliotheksnutzer zu offerieren. Letztlich sind teilautomatisierte Empfehlungssysteme Wegbereiter, in einer informationsdurchfluteten Welt Wissen effizienter zu akquirieren.

Kracht definiert Wissen als Verhaltensposition, die auch beispielsweise bei Büchern oder der Kommunikation zwischen Servern vorkommt [98]. Wissen entsteht nach dieser Betrachtungsweise mittels der Informationsverarbeitung durch Anwendung von Pragmatik oder einer Vernetzung von Erfahrungen entsteht. Dies wird häufig auch in der Wissenspyramide nach Aamodt und Nygård zusammengefasst [4], vergleiche hierzu auch [67].

Mit steigenden Verfügbarkeiten und leichterem Zugang zu elektronischen Publikationen kann argumentiert werden, dass Empfehlungssysteme diese ebenso berücksichtigen müssen. Dies ist aus mehreren Gründen problematisch. Einerseits muss dabei eine Heterogenität (z.B. Datenformate, Systemumgebungen oder bereitgestellter Inhalt) überwunden werden, denn die eigentlichen Daten entstehen in den Verlagen und andererseits ist eine Vollständigkeit aufgrund der Bereitstellung in digitalen Semesterapparaten (z.B durch Urheberrecht oder Ressourceneinsatz) nicht gewährleistet. Gedruckte Bücher bieten aus lernpsychologischer Sicht Vorteile beim Wissenserwerb.

Dies wird unter anderem deutlich in der Studie von Kerr und Symons, in der die Wiedergabeleistung von Text bei gedruckten Materialien höher war, als bei den elektronischen Versionen [90]. Auch die Autoren in [106] konnten in ihrer Studie zeigen, dass im Vergleich zwischen gedrucktem und elektronischem Buch, der Wissenszuwachs wesentlich größer ist, wenn das gedruckte Buch gelesen wird. Auch die Studie von Singer und Alexander kommt zu einem ähnlichen Schluss. Obwohl die Kernidee der Texte bei beiden Medien von den Lesern gleich gut identifiziert wird, werden wichtige Punkte zur Kernidee und andere relevante Information bei den Lesern gedruckter Materialien signifikant besser wiedergegeben [138]. In dieser Arbeit wird die Meinung vertreten, dass ausleihbare Bücher einen wichtigen Kontext in wissenschaftlichen Bibliotheken bilden und zudem das gedruckte Buch auch in Zukunft einen essenziellen Beitrag in der Wissensakquisition darstellt.

1.1 Motivation: Die UB Magdeburg

An der Universitätsbibliothek Magdeburg, die durch die Otto-von-Guericke-Universität vorwiegend durch ein technisches Profil geprägt ist, existieren mit Stand Jahresende 2017 ca. 1.220.000 ausleihba-

[1] Aus Gründen der leichteren Lesbarkeit wird in der vorliegenden Arbeit die gewohnte männliche Sprachform bei personenbezogenen Substantiven und Pronomen verwendet. Dies impliziert jedoch keine Benachteiligung anderer Geschlechter, sondern soll im Sinne der sprachlichen Vereinfachung als geschlechtsneutral zu verstehen sein.

re Exemplare. Diese können durch den Nutzer sowohl im Online Public Access Catalog (OPAC) wie auch dem Discovery-Dienst UBfind gefunden, reserviert oder bestellt werden. Während der OPAC für die gezielte Suche im Bestand genutzt wird und im Jahr 2017 ca. 329.000 Anfragen erfolgten, ermöglicht das Discovery-System eine breite Ergebnisraumbetrachtung. Dies liegt am verwendeten Vektorraummodell und der Verwendung von Ähnlichkeitssuchen. Im Jahr 2017 wurden etwa 651.000 Anfragen an UBfind gestellt. Beide Systeme verwenden keinen auf dem Nutzerverhalten basierenden Algorithmus zur Identifikation von ähnlichen Produkten und geben in diesem Sinn keine Empfehlung bei der Nutzersuche.

Aufgrund der Systemgegebenheiten soll ein flexibles Empfehlungssystem entwickelt werden, dass sich in beide Umgebungen einbetten lässt und zugleich das Profil der Otto-von-Guericke-Universität Magdeburg berücksichtigt. Somit können die Empfehlungen nicht einfach aus dem Gemeinsamen Virtuellen Katalog (GVK) ermittelt werden. Vielmehr sollen die für den Ausleihprozess verfügbaren Informationen als Basis des Empfehlungssystems genutzt werden. Dies bedeutet aber auch, dass nicht für alle im Bestand verfügbaren ausleihbaren Titel Empfehlungen ausgesprochen werden können. Dies betrifft sowohl „Ladenhüter“ wie auch frisch eingetroffene Titel. An dieser Stelle müssen für neu-aufgenommene Titel dann weiterhin die bewährten Verfahren, wie z.B. die „Neuerwerbungsausstellung“, eingesetzt werden.

Ausleihen von Bestandsexemplaren erfolgen im Lokalen Bibliothekssystem (LBS). Dabei werden Prozess- wie auch rechtliche Vorgaben berücksichtigt. Im Sinne der Datensparsamkeit werden nur die Daten erhoben und verarbeitet, die für die Ausleihe notwendig sind. Dies stellt jedoch eine Hürde für einen Data Mining Ansatz dar, da eine notwendige Historisierung der Ausleihdaten nicht erfolgt. Dies muss durch das Empfehlungssystem geleistet werden. Zugleich gelten die Vorgaben auch für dieses Informationssystem. Daher dürfen personenbezogene Daten, für die das Einverständnis zur systembezogenen Datenverarbeitung nicht vorliegt, nicht erhoben oder verarbeitet werden.

Eine weitere gesetzte Anforderung an das Empfehlungssystem ist die Güte der Empfehlungen. Mit fortschreitender Zeit kann es in den Fachdisziplinen dazu kommen, dass früher häufig genutzte Titel nicht mehr relevant sind. Insbesondere in technischen Disziplinen ist die wissenschaftliche Entwicklung derart rasant, dass eine zeitliche Betrachtung relevant ist. In den Geisteswissenschaften ist dieser Umstand weniger wichtig.

1.2 Forschungsfragen

Empfehlungssysteme existieren im kommerziellen Bereich schon seit vielen Jahren und sind ein entscheidender Erfolgsfaktor für Online-Plattformen. Eine Übertragung auf Katalogsysteme wissenschaftlicher Bibliotheken ist beispielsweise mit Bibtip [117] erhältlich. Jedoch werden dabei nicht die ausgeliehenen Bücher berücksichtigt, sondern es erfolgt eine Empfehlung auf Basis des Suchverhaltens im Katalog. Dabei erfolgt die Annahme, dass die Menge an Transaktionsdaten wesentlich höher ist und somit statistische Algorithmen bessere Empfehlungen liefern als über das Ausleihsystem. Mit der verstärkten Durchdringung von Discoverydiensten existieren heute aber weitere Recherchetools, die in diesem Ansatz ausgeschlossen werden bzw. zunächst zusammengeführt werden müssen. Ein weiteres Argument ist der Datenschutz, der bei personenbezogenen Daten berücksichtigt werden muss. Die Argumentation, das es durch Empfehlungen auf Ausleihbasis zu einem Ausschluss von Exemplaren kommt, die sich im Präsenzbestand befinden und somit nicht ausgeliehen werden können, kann entkräftet werden. Heutzutage existieren neben dem Präsenzexemplar weitere Exemplare des Titels, die sehr wohl im Ausleihsystem erfasst werden und es somit auf Titelebene zu Daten und damit Empfehlungen kommen kann.

Empfehlungssysteme arbeiten auf unterschiedlichen Datenbeständen und dementsprechenden Kontextdaten. So können sowohl Nutzerbewertungen als auch verhaltensbasierte Daten in einem Empfehlungssystem verwendet werden. Im deutschsprachigen Raum existieren für wissenschaftliche Bibliotheken beispielsweise Empfehlungssysteme für das Suchverhalten in den Katalogen [116], jedoch werden die eigentlichen Ausleihen nicht berücksichtigt. Diese Arbeit versucht diese Lücke zu schließen, denn Buchempfehlungen basierend auf den Ausleihdaten besitzen einen anderen Kontext als die Empfehlungen basierend auf dem Suchverhalten. Letztlich steht hinter diesem Ansatz die Idee, dass dem Nutzer eine Empfehlung für die Ausleihe angeboten wird. Die Trennung von Nutzerdaten und Empfehlungen spielt aufgrund der rechtlichen Vorgaben eine zentrale Rolle. Eine damit einhergehende Auseinandersetzung im Themengebiet Datensicherheit ist daher ein wesentlicher Beitrag der vorliegenden Arbeit.

Zusätzlich sind die Daten über Bücher hierarchisch organisiert. Von der Ebene der Exemplare können die Titelebene oder auch die Werkebene abgeleitet werden. Aber auch die Autorenschaft oder die Publikationsebene sind als weitere Aggregationsstufe möglich. Für diese unterschiedlichen Ebenen ergeben sich ebenfalls differenzierte Empfehlungen. So kann auf Titelebene jedes Exemplar empfohlen werden, dass dem Titel zugeschrieben wird, da dem Nutzer kein unmittelbarer Unterschied im Sinne der Ausleihe vorliegt. Auf Werkebene kann es im Gegensatz schon wichtiger sein, dass beispielsweise eine spezielle Auflage empfohlen werden sollte (meist die aktuellste). Aber auch auf der Ebene der geistigen Schöpfer ändert sich die Empfehlung, da an dieser Stelle alle Bücher eines Autors zusammengefasst werden und beispielsweise Goethes Werke als Empfehlung Schillers Werke aufweisen können. Das Granularitätslevel wird dabei durch den Nutzer und seine Intention festgesetzt. In Konsequenz versucht diese Arbeit zu ermitteln, welches Level innerhalb der Empfehlungsobjekte geeignet ist.

Mit fortschreitender Zeit verändern sich nicht nur Nutzergewohnheiten, sondern wissenschaftliche Bücher werden nicht mehr unmittelbar empfohlen, z.B. durch Dozenten in den Veranstaltungen oder aber einer Änderung in der Aufstellung des Semesterapparats, bzw. durch neuere Auflagen ersetzt. Für die Empfehlung stellt sich dabei die Frage, ob ältere Informationen genauso genutzt werden sollten, wie die aktuell vorliegenden. Im wissenschaftlichen Umfeld existiert eine hohe Dynamik, die sich insbesondere in den Publikationen widerspiegelt. Daher wäre es wünschenswert, wenn früher stark nachgefragte Werke, die heute im Regal verstauben, ebenfalls im Empfehlungssystem einen geringeren Stellenwert erhalten. Momentan unbeantwortet ist die Frage, wie sich ältere Informationen der Ausleihe im Empfehlungssystem abbilden lassen und daher widmet sich diese Arbeit dem Kontext des Vergessens „veralteter“ Ausleihinformationen.

Das Ausleihsystem dient als operative Anwendung und sollte daher nicht durch analytische Komponenten blockiert bzw. eingeschränkt werden [97]. Auch eine Änderung am Datenmodell oder die Verwendung von datenbankspezifischen Techniken wie Triggern ist nicht zielführend. Daher soll das Empfehlungssystem modular aufgebaut sein und sich mittels Schnittstellen in die Systemlandschaft integrieren. Aufgrund der Anforderungen im Bibliothekskontext sind für ein Empfehlungssystem auf Basis des Ausleihsystems die Frage nach der Data-Privacy-konformen Gestaltung des Warenkorbes, dem Level auf dem die Elemente des Warenkorbes betrachtet werden und wie zeitlich begrenzt nutzbare Profile genutzt werden können von Relevanz.

Die Erstellung und Gestaltung der Warenkörbe für ein Empfehlungssystem aus Buchausleihdaten ist zentraler Punkt der vorliegenden Arbeit. Diese Arbeit widmet sich dabei den folgenden drei Forschungsfragen.

Forschungsfrage 1 Wie können Data-Privacy-Anforderungen für ein Buchausleih-Empfehlungssystem umgesetzt werden?

Der Einsatz von Datenschutzmaßnahmen kann durch Anonymisierungs- oder Pseudonymisierungsverfahren durchgeführt werden. Dies bedeutet, dass bereits vor dem Data Mining ein Aufwand betrieben wird. Aber auch während des Data Mining sind durch geeignete Wahl von Parametern Aspekte des Datenschutzes inkludierbar. Letztlich lassen sich auch bei der Ergebnispräsentation Techniken anwenden, die einen Rückschluss auf das Individuum erschweren bzw. unmöglich machen. Diese Arbeit eruiert daher alle drei Ebenen.

Forschungsfrage 2 Wie lassen sich hierarchische Strukturen für die Warenkorbanalyse sinnvoll einsetzen?

Im Bibliotheksumfeld sind Hierarchien und Ontologien ein häufig eingesetztes Mittel. So ist von der Ebene der Exemplare auf Titel- und Werkebene eine Navigation möglich. Dies kann aber auch z.B. auf Autorenebene, Verlagsangebotsebene oder Paketebene transferiert werden. Aber auch auf der Ebene der Nutzer wird in Bibliotheken häufig auf Gruppenbildungsmöglichkeiten zurückgegriffen, z.B. Nutzergruppen oder bei Studierenden die Fachdisziplin. Das Empfehlungssystem soll diese Hierarchien abbilden können und geeignet unterstützen. Dabei muss zugleich berücksichtigt werden, dass die Ergebnisrepräsentation diesbezüglich auch zu einem stark unterschiedlichen Interpretationsraum wird.

Forschungsfrage 3 Welche Strategie zum „Vergessen“ älterer Ausleihen führt zu ausreichenden Empfehlungen?

Aufgrund der hohen Anzahl an Produkten, insbesondere auf Ebene der Exemplare und der vergleichsweise geringen Anzahl an Ausleihen sind für die Güte des Data Mining Verfahrens eine möglichst große Anzahl an Transaktionen notwendig. Ein Vergessen älterer Transaktionen führt dabei auch zu einem Informationsverlust für das Verfahren. Jedoch stellt sich die Frage, ob durch einem dem Simulated Annealing angelehnten Verfahren hinreichend gute Empfehlungen unter Berücksichtigung des Wandels der Bibliotheksprodukte gewährleistet werden kann.

1.3 Aufbau der Arbeit

Die Arbeit ist wie folgt strukturiert. In Kapitel 2 werden die für diese Arbeit notwendigen Begriffe und Techniken vorgestellt. Dabei wird insbesondere auf Empfehlungssysteme eingegangen. Zusätzlich müssen im produktiven Betrieb von Informationssystemen rechtliche Belange beachtet werden. Hierbei spielen insbesondere Fragen des Datenschutzes und der Datensicherheit eine Rolle. Auf den ersten Aspekt wird dabei im Rahmen dieser Arbeit eingegangen, die Datensicherheit sollte weitestgehend im Konzept des Systems berücksichtigt werden und spielt daher für diese Arbeit eine untergeordnete Rolle. Jedoch sind auch Datenhaltungskonzepte notwendig, da für den Systembetrieb mehrere unterschiedliche Systeme gekoppelt werden müssen. In der Literatur aber auch im produktiven Einsatz existieren bereits Empfehlungssysteme, die einen ähnlichen Kontext aufweisen. Daher wird kurz auf ausgewählte Arbeiten eingegangen.

In Kapitel 3 wird das Konzept für das Empfehlungssystem auf Basis der Warenkorbanalyse vorgestellt. Hierbei werden die im vorangegangenen Kapitel gelegten Grundlagen für einzelne

Belange des Empfehlungssystems näher betrachtet. Details der algorithmischen Implementierung, die sowohl aus der Literatur stammen als auch die beiden Zielsysteme betreffen, werden präsentiert.

Die eigentliche Umsetzung inklusive der Implementierungsfragen wird hingegen in Kapitel 4 betrachtet. Hier sind neben der Warenkorbanalyse insbesondere die Transformationen im Fokus. Auch das detaillierte Vorgehen wird näher beschrieben, so dass ein Data-Privacy-By-Design-Konzept unmittelbar ersichtlich wird. Letztlich wird beispielhaft am Katalogsystem die Einbettung der Empfehlungen vorgestellt.

Eine Evaluation erfolgt in Kapitel 5. Hierbei werden zwei Datensätze genutzt, ein künstlich erzeugter und ein zweiter aus dem Ausleihsystem der UB Magdeburg. Fragen hinsichtlich der Parametrisierung werden untersucht und die Ergebnisse inklusive der Ergebnisse zu den Forschungsfragen präsentiert. Kapitel 6 schließt die Arbeit und gibt einen Ausblick auf zukünftige Weiterentwicklungen.

2 Grundlagen

In diesem Abschnitt werden die für die vorliegende Arbeit notwendigen Grundlagen dargestellt. Neben den Verfahren zur Datenhaltung im LBS wird dabei insbesondere auf die Verfahren im Data Mining zur Warenkorbanalyse eingegangen. Ein wichtiger zu berücksichtigender Aspekt spielt dabei Data Privacy. Hierzu müssen deutsche Datenschutzgesetze beachtet werden, z.B. das Bundesdatenschutzgesetz [3] oder das Datenschutzgesetz des Landes Sachsen-Anhalt [1] oder die ab 25. Mai 2018 in Kraft tretende Datenschutzgrundverordnung [2]. Zudem wird in Abschnitt 2.5 auf exemplarische Empfehlungssysteme für wissenschaftliche Bibliotheken eingegangen.

2.1 Datenhaltung

Die Datenhaltung erfolgt in Systemen in sehr unterschiedlicher Art und Weise. Für digitale Inhalte haben sich zunächst unterschiedliche Dateisysteme etabliert. Seit den 60er Jahren haben sich bereits Datenbanken als Datenhaltungssysteme entwickelt. Dabei ist das relationale Datenhaltungssystem auch heute noch das am häufigsten genutzte [133]. Dies liegt einerseits an der schnellen Verbreitung der Systeme und andererseits am Reifegrad und der damit einhergehenden starken Durchdringung sowohl im betrieblichen wie auch privaten Umfeld sowie der engen Verzahnung von Forschung und Lehre. Darüber hinaus haben sich Anforderungen siehe z.B. [48] entwickelt, die z.B. durch analytische Systeme [97], Big Data [63] aber auch andere komplexe Datenhaltungen wie Extensible Markup Language (XML) [34] berücksichtigt werden.

Um eine schlanke und zugleich effiziente Datenhaltung zu ermöglichen, müssen sowohl die Quellsysteme berücksichtigt werden als auch Anforderungen an Datensparsamkeit und Datenschutz. Daher ist es einerseits notwendig, über standardisierte Sprachen wie z.B. SQL [85] Zugang zu den Daten zu erhalten. Über bereitgestellte Schnittstellen kann mittels Programmiersprachen wie z.B. Java und PHP eine Datentransformation durchgeführt werden, wobei die daran anschließende Datenanalyse ebenfalls mit Programmierumgebungen wie PHP und R erfolgen kann. In diesem Zusammenhang sind zwar Konzepte aus dem relationalen Datenbanksystem ebenfalls umsetzbar wie z.B. durch Objektrelationale Abbildungen [132] jedoch sollte eine Datenhaltung in einfachen Comma-Separated-Values (CSV)-Dateien ebenso möglich sein.

Für die vorliegende Arbeit wird sowohl auf relationale Datenbanken wie auch auf Dateien zurückgegriffen. Für die Datenhaltung wird auf Dateiebene vorwiegend auf Textformate gesetzt, wobei insbesondere das CSV-Format eine Rolle spielt. Da die Datenhaltung der Ursprungsquellen wie auch das Zielsystem für die Präsentationsebene auf Datenbanken beruhen, werden die wesentlichen Eigenschaften vorgestellt.

2.1.1 Dateiorganisation

Dateien können sowohl unterschiedlich codiert als auch strukturiert vorliegen. An dieser Stelle wird angenommen, dass die Datenbasis als Unicode Transformation Format (UTF)-Codierung vorliegt. Prinzipiell gibt es auch noch weitere Codierungen, wobei American Standard Code for Information

Interchange (ASCII) mit dem 7-Bit Zeichenumfang für die US-Codierungen als Grundlage für American National Standards Institute (ANSI) gilt. Aufgrund der eher technischen Ebene der Kodierung, wird diese nicht weiter betrachtet. Prinzipiell gilt aber, dass eine Transformation von ASCII (bereits seit 1963 standardisiert) oder ANSI zu UTF möglich ist. Daher wird im Folgenden davon ausgegangen, dass die Dateien in UTF-8 vorliegen.

Strukturierte Daten können in sehr unterschiedlichen Formaten abgelegt werden. Zu den momentan häufig im Einsatz befindlichen wird an dieser Stelle eine kurze Aussage getroffen, wobei weder ein Anspruch auf Vollständigkeit erhoben, noch eine detaillierte Übersicht der Vor- und Nachteile vorgenommen wird.

Das CSV-Format stellt ein einfaches Textformat dar, dass durch einen definierten Trenner die einzelnen Attribute abgrenzt. Jede Zeile entspricht hierbei einem Datensatz. Optional kann in der ersten Zeile die Definition der Metadaten erfolgen. Zu den typischen Trennzeichen gehören das Komma, das Semikolon, der Doppelpunkt oder der Tabulator. Im letzten Fall spricht man dann auch vom Tabular-Separated-Values (TSV)-Format. Eine weitere Möglichkeit ist es, nach definierten Abständen (Längenangabe der Zeichen) die Trennung der einzelnen Attribute vorzunehmen. Letztlich ist das CSV-Format eine einfache Lösung, um eine tabellarische Datenstruktur abzulegen, die Text- und Zahleninformationen abbilden kann. Im Kontext der Zeichendarstellung muss jedoch darauf geachtet werden, dass das Trennzeichen für die Attribute nicht mehr zur Verfügung steht. So ist die die Verwendung des Kommazeichens für Gleitkommazahlen nicht möglich, falls es sich um eine CSV-Datei handelt, bei der das Komma als Trennzeichen genutzt wird. Eine weitere Restriktion ist, dass die Metadatenbeschreibung darin besteht, den einzelnen Attributen eine Beschreibung zu verleihen. Darüber hinaus wichtige Metadaten wie z.B. Datentyp oder Wertebereich sowie Zusammenhänge zwischen den Attributen sind hiermit nicht abbildbar.

Bei der XML handelt es sich um eine Auszeichnungssprache, die im Textformat als UTF-8 kodiert gespeichert wird. Hauptziel des Formates ist es, sowohl für den Menschen wie auch den Computer lesbar zu sein. XML ist als Standard des W3C verabschiedet. Es folgt den Designprinzipien der Einfachheit und Allgemeinheit sowie der Nutzung über das Internet. Während ein Textdokument weitestgehend ohne Metainformationen durch den Menschen erfasst werden kann, dient XML im Kontext der dokumentenzentrierten Strukturierung zur Identifikation von einzelnen Abschnitten im Dokument. Bei der datenzentrierten Strukturierung steht die maschinelle Verarbeitung im Vordergrund. Hierbei werden Entitäten eines Datenmodells mit den entsprechenden Attributen und Beziehungen beschrieben. Der Strukturierungsgrad ist sehr hoch. Überdies sind zwischen den beiden Arten auch Mischformen möglich.

Ein weiteres für den Austausch konzipiertes kompaktes Format stellt JavaScript Object Notation (JSON) dar. Kernidee des Formats ist, dass es sich bei jeder Datei um Javascript handelt, die mittels der Methode *eval*() interpretierbar ist. Zugleich ist JSON unabhängig von der Programmiersprache, da Parser in vielen anderen Sprachen neben Javascript verfügbar sind. Mit JSON wird im Vergleich zu XML eine größere Flexibilität bei den APIs erreicht und zugleich ein kleineres Austauschformat genutzt, jedoch stellt letzteres Format bei einer rigiden Schnittstelle die bessere Wahl dar.

Eine weitere Alternative stellen Dateien im Webstandard Resource Description Framework (RDF) dar [107]. Dieser dient vor allem der Beschreibung von beliebigen Dingen (Ressourcen) und deren Zusammenhängen. Ziel des Standards ist ein Austausch von Informationen im Web. Dabei wird auf eine wohldefinierte formale Semantik gesetzt, indem ein Tripel bestehend aus Subjekt, Prädikat und Objekt als Aussage zu den Ressourcen genutzt wird. Obwohl RDF unabhängig einer Notation ist, wird häufig XML eingesetzt. Auch die Metadaten können direkt in RDF beschrieben werden. Eine Übersicht sowie detaillierte Informationen zu RDF, XML und anderen Webstandards findet

sich in [82].

Zwischen den einzelnen Datenformaten kann mittels Transformationen gewechselt werden. Da jedoch nicht alle Informationen im gleichen Umfang abbildbar sind, müssen Informationsverluste in Kauf genommen werden. Häufig wird das Format gewählt, dass für den Einsatzzweck entsprechend gut nutzbar ist. Mit steigendem Umfang der Datenmenge bzw. einer Homogenität der Daten kann es sinnvoll sein, ein einfacheres Format zu nutzen, um die Datenverarbeitung effizient zu gestalten.

2.1.2 Datenbankmanagementsysteme

Ein Datenbankmanagementsystem verwaltet die als Basis dienende Datenbank und ist unter anderem verantwortlich für Authentifizierung, Datenverwaltung, Datenmodell und Absicherung der Anforderungen sowie die Funktionalität des Systems. Ein Datenbankmanagementsystem (DBMS) ist die Software, die die Datenbank als Datenspeicher umfassend verwaltet. Ein weit verbreiteter Typ sind relationale DBMSe. Diese basieren auf dem relationalen Modell nach Codd [47]. Für die Datenverwaltung und die Datenabfrage bildet die Structured Query Language (SQL) eine wichtige Grundlage, da darüber sowohl die Zugriffskontrolle, die Datenmanipulation als die Datendefinition möglich sind.

Das relationale Modell organisiert Daten spalten- und zeilenweise in Tabellen, die häufig auch als Relation bezeichnet werden. Im klassischen zeilenbasierten DBMS besteht ein Datensatz aus unterschiedlichen Attributen, die in den Spalten abgelegt sind. Der eigentliche Datensatz, auch als Tupel bezeichnet, ist in einer Zeile hinterlegt. Jede Zeile in einer Relation kann eindeutig durch den Schlüssel identifiziert werden, d.h. in einer Relation ist jede Zeilenausprägung individuell. Die Verknüpfung von unterschiedlichen Tupeln ist durch einen Schlüsselverbund über verschiedene Tabellen möglich. Diese Schlüssel werden als Fremdschlüssel bezeichnet. Im Fall einer Verknüpfung von Tabellen wird von einer Relationship gesprochen. Es ist auch möglich, dass die Datenablage in spaltenorientierter Form erfolgt [5], dabei steht dann eine effiziente Verarbeitung im Sinne der Verarbeitung der Attribute im Vordergrund.

Ein wichtiges Konzept für die effiziente und ordentliche Datenverarbeitung ist ACID [71]. Dieses steht für Atomarität, Konsistenz, Isolation und Dauerhaftigkeit.

- *Atomarität* bedeutet, dass in einer Reihe von Datenbankoperationen entweder alle oder keine ausgeführt werden. Dies führt dazu, dass das Datenbanksystem sich stets in einem konsistenten Zustand befindet. Auf der anderen Seite heißt dies auch, dass während die Operationen noch ausgeführt werden, der Datenbankzustand nach außen unverändert ist, so als wenn keine Operation ausgeführt würde. Erst nach Abschluss aller Operationen befindet sich die Datenbank in einem neuen Zustand.

- *Konsistenz* garantiert, dass zukünftige Transaktionen Zugriff auf die Effekte anderer bereits eingecheckter Transaktionen haben. Dies bedeutet auch, dass keine Restriktionen verletzt werden. Somit werden die Operationen einer Transaktion sorgfältig, korrekt und gültig hinsichtlich der Anwendungssemantik ausgeführt.

- *Isolation* legt fest, wie eine Transaktion für andere Nutzer oder Systeme sichtbar ist. Mit einem geringeren Isolationslevel können mehr Nutzer auf die gleichen Daten zur gleichen Zeit zugreifen. Dies geht dann einher mit einer größeren Rate an Nebenläufigkeitsanomalien. Mit einem höheren Isolationslevel werden diese Effekte verringert, dies erfolgt aber zu Lasten der benötigten Ressourcen und führt zur Blockade einiger Transaktionen.

- *Dauerhaftigkeit* garantiert, dass eine einmal eingecheckte Transaktion fortbesteht. Dies wird beispielsweise auch für einen Systemcrash garantiert.

Transaktionale Systeme Das Paradigma Online Transaction Processing (OLTP) steht für die direkte und prompte Ausführung im transaktionalen Kontext. Eine Vielzahl von Nutzern und Systemen kann auf einen großen Datenbestand zugreifen, diesen verändern, anreichern und Teile löschen. Im Fokus steht die Bearbeitung einzelner Datensätze.

Der Erfolg der relationalen Datenbank ist nicht nur dem Transaktionskonzept zuzuschreiben, sondern insbesondere der Tatsache, dass ein Auffinden im Datenbestand mit sub-linearen Rechenaufwand möglich ist. Hierzu werden Indexstrukturen eingesetzt, wobei insbesondere der von Bayer und McCreight vorgeschlagene B-Baum als wichtiges Instrument zu nennen ist [19]. Neben der eindimensionalen Repräsentation sind auch multidimensionale Indexstrukturen möglich, die mehrere Attribute gleichzeitig berücksichtigen. Insbesondere im Online Analytical Processing (OLAP) sind die Anforderungen hinsichtlich multidimensionaler Anfragen zu berücksichtigen.

Analytische Systeme Für das Data Mining sind neben den relationalen Datenbankbeständen häufig auch zusätzliche Daten notwendig, wie z.B. historische Daten für Zeitreihenanalysen. Daher wurde in den 90er Jahren von Inmon das Konzept der Data Warehouses eingeführt [84], vgl. hierzu auch [97]. Aufgrund der Komplexität der Anwendungsszenarien sind die Entwicklungen mannigfaltig, von Indexstrukturen für Geoinformationssysteme, wie z.B. der R-Baum-Familie [73], über Hauptspeicherdatenbanken [126] bis hin zu speziellen multidimensionalen Datenstrukturen, wie z.B. Dwarf [139] und Elf [36]. Letztlich kommen aber auch Dateiformate wieder zum Einsatz, um die Flexibilität im Data Mining zu gewähren.

2.2 Data Privacy

Mit der Durchdringung von Suchmaschinen und Data Mining Verfahren können personenbezogene Daten gesammelt und mit einfach handhabbaren Methoden analysiert werden. Aber auch die Anwendung von Big Data Methoden ermöglicht es, bei nicht geprüften Einzeldaten auf unbekannte Zusammenhänge zu schließen, die allein aufgrund der riesigen Datenmengen und der identifizierten Korrelationen zu einem Wahrheitsgehalt führen [16, 105]. Gesetze und Regeln versuchen insbesondere personenbezogene Daten zu schützen. Der deutsche Begriff Datenschutz umfasst dabei die Begriffe Security und Privacy. Unter Security werden alle Aufgaben zusammengeführt, die verhindern sollen, dass eine Einschränkung der Vertraulichkeit, eine Manipulation oder Störung sowie Beschränkung der Verfügbarkeit der technischen Geräte von außen erfolgt.

Unter Privacy wird insbesondere das Recht verstanden, frei von geheimer Überwachung zu sein und jederzeit über die persönlichen oder organisatorischen Informationen und ihre Weitergabe zu bestimmen. Personenbezogene Daten stehen besonders im Fokus, wenn es um Data Privacy geht. Auch hierbei erfolgt eine weitere Unterteilung, in sensible und vertrauliche Daten. Während der letztere Begriff unter anderem auch vertraglich geregelt sein kann, definiert §3 Absatz 9 BDSG sensible Daten als „Angaben über die rassische und ethnische Herkunft, politische Meinungen, religiöse oder philosophische Überzeugungen, Gewerkschaftszugehörigkeit, Gesundheit oder Sexualleben“ [3]. Auch diese Daten dürfen erhoben werden, obliegen aber einer erschwerten Verarbeitung.

Darüber hinaus werden unter Datensicherheit technische und organisatorische Maßnahmen gefasst, die Bedrohungen wie Manipulation oder unberechtigte Kenntnisnahme verstehen, vergleiche hierzu beispielsweise §9 BDSG [3]. Um diese Möglichkeit der Rückverfolgung zu umgehen, ist es notwendig,

dem Grundsatz der Datensparsamkeit strikt zu folgen. Denn so werden nur Attribute gespeichert, die wirklich benötigt werden. Zudem kann es hilfreich sein, die Datenmenge, das heißt die Anzahl der Fälle, so groß wie möglich zu halten. Dies würde die Selektivität der Attribute reduzieren.

Anforderungen an Datensicherheit werden häufig in drei Bereiche unterteilt [27]:

- *Vertraulichkeit*: Gegenüber Unbefugten werden die Daten nicht verfügbar gemacht.
- *Integrität*: Eine Prüfung von Datenänderungen ist möglich und somit werden Änderungen der Daten erkannt.
- *Verfügbarkeit*: Die Daten werden zum Nutzungszeitpunkt uneingeschränkt bereitgestellt.

Dittmann sieht acht Sicherheitsanforderungen als wichtig an, wobei die oben genannte Anforderung der Verfügbarkeit nicht betrachtet wird [55]:

- Zugriffskontrolle,
- Authentizität,
- Vertraulichkeit,
- Integrität,
- Verbindlichkeit,
- Nicht-Abstreitbarkeit,
- Urheberrechte und
- Persönlichkeitsschutz.

Für diese Anforderungen existieren unterschiedliche Managementkonzepte, wobei die relevanten sich in die Gruppe der Kryptologie einstufen lassen [55]. Somit ist die Wissenschaft von der Geheimhaltung von Informationen und ihren zugeordneten Algorithmen an dieser Stelle genauer zu betrachten. Nicht alle Anforderungen sind für den Einsatz im Empfehlungssystem dabei notwendig. Zudem erfüllen nicht alle anonymisierenden Verfahren alle Anforderungen gleichzeitig.

Das Bundesdatenschutzgesetz definiert in §3 personenbezogene Daten als Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person [3]. Daher müssen Maßnahmen getroffen werden, diese Daten zu schützen. Denn die Person kann aufgrund ihrer informationellen Selbstbestimmung festlegen, wer über welche persönlichen Informationen verfügen darf. Bei dem Begriff der bestimmbaren Person handelt es sich um einen unscharfen Rechtsbegriff. Dabei muss betrachtet werden von wem und mit wie viel Aufwand sich die Bestimmung durchführen lässt.

Eine wichtige Anforderung an das Empfehlungssystem ist es damit, dass personenbezogene Daten nicht verwendet bzw. ein Rückschluss auf eine Person nicht möglich ist. Als Ausgangsbasis für das Empfehlungssystem stehen personenbezogene Daten in der Verwendung durch das Ausleihsystem. Somit muss gewährleistet werden, dass die entsprechenden Daten konform mit den Regeln aufbereitet werden, damit kein Personenbezug mehr möglich ist. Dies erfolgt durch Datenanonymisierung. Aber auch ein Rückschluss der Ergebnisse der Präsentationsebene auf die Einzelperson muss verhindert werden. An dieser Stelle spielen unterschiedliche Strategien eine Rolle, die in Abschnitt 2.2.2 näher beschrieben werden.

2.2.1 Datenanonymisierung

Die Anonymisierung von Daten muss so erfolgen, dass sich Personen nicht unmittelbar identifizieren lassen und sich aus den Daten auch keine Personen ableiten lassen.

In der Literatur, aber auch in Regeln und Gesetzen, werden oftmals die Begrifflichkeiten Anonymisierung und Pseudonymisierung verwendet. Während bei der Anonymisierung die Daten so verändert werden, dass Angaben über die Person oder Sache nicht oder zumindest nur mit einem unverhältnismäßig hohen Aufwand an Zeit und Ressourcen einer Person zugeordnet werden können, erfolgt bei der Pseudonymisierung die Ersetzung von Identifikationsmerkmalen durch Kennzeichen, so dass eine unmittelbare Bestimmung der Personen wesentlich erschwert ist [3].

Problematisch ist hierbei die Rückidentifikation für einen Anwenderkreis, so lange die Ersetzungsmerkmale entweder mit den personenbezogenen Daten gemeinsam vorgehalten werden oder der Rückschluss ohne große Umstände erfolgen kann. In diesem Sinne muss nach den rechtlichen Vorgaben davon ausgegangen werden, dass weiterhin personenbezogene Daten vorliegen.

In der Praxis existieren unterschiedliche Gründe für die Weitergabe und Verarbeitung personenbezogener Daten. So können Statistiken oder Forschungsanliegen aber auch Gesetzesanforderungen oder Dienstverbesserungen Gründe für die Weiterverarbeitung sein. In diesen Fällen müssen Anforderungen an die Anonymisierung der Daten getroffen werden.

Die Anonymisierung kann auf sehr unterschiedlichen Ebenen und mit unterschiedlichen Verfahren erreicht werden. Einerseits ist ein Hinzufügen von Rauschen auf die Daten möglich. Dies bedeutet, dass jeder Datensatz verändert wird und somit weniger Rückschluss auf die wahren Werte erfolgen kann. Dabei müssen jedoch auch die Datendomänen wie auch die Varianz des Rauschens beachtet werden. Für die weitere Analyse kann dies dann ggfs. zu Missinterpretationen führen. Dies hat jedoch auch zur Folge, dass damit für das Buchausleih-Empfehlungssystem keine verlässliche Datenbasis mehr vorliegt. Andererseits können zusätzlich Datensätze hinzugefügt werden oder in den einzelnen Datensätzen werden Attribute unterdrückt.

Darüber hinaus lassen sich zwei weitere Ansätze zur Anonymisierung identifizieren: Die Änderung der Reihenfolge der Datensätze kann unter Umständen bereits ausreichen, eine Re-Identifikation unmöglich zu machen. Mit der Aggregation bzw. Verdichtung der Daten oder der Darstellung der Wertebereiche ergibt sich eine weitere Möglichkeit, den Personenbezug nicht mehr herstellen zu können.

2.2.2 Vermeidung von Datenrückschlüssen

Das Weglassen von identifizierenden Attributen bzw. den unmittelbaren Personenidentifizierern reicht nicht aus, um die Anonymität zu sichern. Denn alternative Kandidatenschlüssel könnten ebenso eingesetzt werden. Hierbei spielen insbesondere zusammengesetzte Attribute als Schlüssel eine Rolle. An dieser Stelle wird das Konzept des Quasi-Identifizierers nach Dalenius vorgestellt [50].

Ein Quasi-Identifizierer Q_T für einen Datensatz in der Datenbasis T besteht aus der Menge der Attribute

$$(A_i, \ldots, A_j) \subseteq (A_1, \ldots, A_n) \text{ mit } \exists p_i \in U : f_g(f_c(p_i)[Q_T]) = p_i,$$

wobei die Datenbasis T n Attribute enthält und die Population U für die Datensätze mit den Funktionen

$$f_c : U \to T \text{ und } f_g : T \to U' \text{ mit } U \subseteq U'$$

definiert ist.

Bereits mit wenigen Attributen lassen sich Rückschlüsse auf unterschiedliche Weise treffen, die

bis zu einer Personenidentifikation führen können. Sweeney hat bereits aufgezeigt, dass in den USA drei Merkmale ausreichen (Postleitzahl, Geschlecht und Geburtsdatum), um aus öffentlich zugänglichen Quellen die Information zur Person in 87% zu erzielen [144].

Zum Schutz der Daten wurde an dieser Stelle die k-Anonymität eingeführt [145]. Diese bedeutet, dass keine Rückschlüsse auf ein Individuum erfolgen können. Somit müssen die Daten derart gestaltet werden, dass der Quasi-Identifizierer in den anonymisierten Daten auf mindestens k Datensätze verweist. k-Anonymität bezieht sich dabei in der Gesamtheit aller Datensätze auf die kleinste gemeinsame Menge der Quasi-Identifizierer. Somit ist jeder Eintrag zumindest nicht differenzierbar zu (k-1) anderen Einträgen in der Datenbasis. Diese Datensätze bilden eine Äquivalenzklasse.

Um k-Anonymität erzielen zu können, müssen die Quasi-Identifizierer beispielsweise so verändert werden, dass sie weniger spezifisch sind. Im Beispiel der Postleitzahlen können so nur die ersten Ziffern genutzt werden. Diese Strategie der Verdichtung von Informationen wird als Generalisierung bezeichnet. Für den Fall, dass die Generalisierung zu einem zu hohen Verdichtungsgrad führt, kann auch ein Löschen der Datensätze vor Veröffentlichung erfolgen. Dies ist insbesondere dann anwendbar, wenn Ausreißer in den Datenbeständen auftreten. Anzumerken ist an dieser Stelle, dass k-Anonymität keine Data Privacy garantieren kann. Wenn etwa in den sensiblen Daten eine zu große Homogenität auftritt oder der Angreifer Hintergrundwissen besitzt, ist es möglich, Rückschlüsse aus den anonymisierten Datenbeständen zu ziehen.

Daher wurde das Konzept der ℓ-Diversität eingeführt [104]. Hierbei wird gefordert, dass die sensitiven Daten in einer Äquivalenzklasse der Quasi-Identifizierer mindestens I unterschiedliche Werte aufweisen. Neben diesem Ansatz gibt es noch Erweiterungen, da Angriffe mit probabilistischen Methoden auch an dieser Stelle zu einer Re-Identifikation führen können. Problematisch ist vor allem, dass für sehr schiefe Datenverteilungen mit hoher Wahrscheinlichkeit ein Rückschluss möglich ist. Aber uninteressante Attributsausprägungen, die z.B. keine sensiblen Informationen aufweisen, benötigen die ℓ-Diversität nicht. Auch das Erreichen einer gesetzten ℓ-Diversität ist schwer zu erreichen, da eine Restriktion der Äquivalenzklassen hiermit häufig einhergeht. In diesem Konzept werden weder die Gesamtverteilung der sensitiven Daten berücksichtigt, noch wird die Semantik der Daten genutzt.

Ein weiterer Ansatz diesbezüglich ist die t-Closeness [101]. Hierbei wird gefordert, dass die Verteilung der sensiblen Daten innerhalb einer Äquivalenzklasse möglichst nahe an der Gesamtverteilung liegen sollte. Aber auch wenn Datensätze alle drei Anforderungen zugleich erfüllen, ist es möglich, mit Hintergrundwissen eine Re-Identifikation zu erreichen. Mit den Vorgehensweisen geht einher, dass die Daten nur syntaktisch anonymisiert werden, ohne dabei die Datenanalyse zu berücksichtigen. Zudem können bei k-Anonymität sensible Informationen möglicherweise re-identifiziert werden. Mittels der Quasi-Identifizierer wird angenommen, dass ein Angreifer kein zusätzliches Hintergrundwissen hat und da die Konzepte auf Datenlokalität beruhen, wird bei Anwendung der Konzepte der Nutzen von realen Daten weitestgehend vernichtet.

Eine hohe Datensicherheit im Sinne der Data Privacy hat auch ein geringen Nutzen für analytische Verfahren zur Folge, da Zusammenhänge entfernt werden bzw. eine Aggregation feingranularer Daten erfolgt. Ein anderer Ansatz basiert auf dem Konzept, die Datendarstellung zu verändern. Dieses Konzept wird Differential Privacy [57] genannt und insbesondere für Aggregationsfunktionen angewandt.

Die Kernidee ist, dass das Weglassen oder Hinzufügen eines Datensatzes keine Informationen bereitstellt. Da keine Informationen durch die Präsenz eines einzelnen Datensatzes aufgedeckt werden, handelt es sich bei der Differential Privacy um eine starke Data Privacy. Man spricht von ϵ-Differential Privacy, wenn ein Zufallsmechanismus Z bei Anwendung auf zwei Datenbestände, die

sich nur durch einen Datensatz unterscheiden und für jedes Anfrageergebnis E gilt:

$$\frac{Pr\left[Z(D_1) \in E\right]}{Pr\left[Z(D_2) \in E\right]} \leq \epsilon.$$

ϵ wird auch als Privacy Budget oder Level bezeichnet.

Um eine Bewertung des beabsichtigten Rauschens vorzunehmen, wird das Konzept der Empfindlichkeit (engl. sensitivity) im Sinne der Rauschfunktion genutzt [58]. Dabei wird in lokale Empfindlichkeit und globale Ebene unterschieden. Die Empfindlichkeit S einer Funktion f ist als der maximale Unterschied definiert, der bei Hinzufügen oder Weglassen eines Datensatzes aus dem Datenbestand D erfolgt:

$$S(f) \geq \left\| f(D') - f(D) \right\|,$$

wobei D' die um einen Datensatz geänderte Datenbasis zu D darstellt. Lokale Empfindlichkeit ist somit definiert als:

$$S_{lokal}\left(f(D)\right) = max \left\| f(D) - f(D') \right\|_1.$$

Die globale Empfindlichkeit S_{global} ergibt sich als das Maximum aller lokalen Empfindlichkeiten. Die Empfindlichkeit hängt somit vom geänderten Element in der Datenbasis und der Rauschfunktion f ab.

Das Verrauschen der Daten im Kontext der differentiellen Privacy erfolgt durch das Hinzufügen von z.B. Laplace oder normal-verteilten Zufallszahlen. Aber auch exponentielle Mechanismen sind möglich. So kann zwischen den Ausgaben der Anfrage nicht mehr im ϵ-Bereich unterschieden werden, sobald gilt [58]:

$$\frac{S_{global}}{\epsilon} \leq Lap\left(\frac{S_{global}}{\epsilon}\right).$$

Wie der Wert für ϵ ermittelt wird, hängt sowohl vom Einsatzgebiet, den Anforderungen und insbesondere dem Datenbestand (inklusive der Metadaten) sowie der Rauschfunktion ab. Eine Schrankenbetrachtung findet sich in [100] oder [114]. In der aktuellen Forschung werden die Mechanismen zum Verrauschen weiter untersucht, siehe z.B. [11] oder [161].

2.3 Recommendersysteme

Empfehlungssysteme (engl. recommender systems) sind spätestens seit dem kommerziellen Erfolg von Amazon[1] beinahe jedem Internetnutzer bekannt [134]. Es wird in der Literatur zwischen kollaborativen und inhaltsbasierten, demografischen und wissensbasierten Techniken für Empfehlungssysteme unterschieden [41]. Darüber hinaus hat sich eine unterschiedlich gestaltete Kombination dieser Ansätze als hybride Form etabliert.

Das Paradigma für Empfehlungssysteme kann als zweiteiliges Problem angesehen werden, welches versucht mittels einer Nutzenfunktion *empfehlung* eine Vorhersage für die Nützlichkeit eines einzelnen Elementes i in der Menge aller Elemente I für einen definierten Nutzer u aus der Menge aller Nutzer U zu machen [86]. Die Nutzenfunktion des Empfehlungsproblems haben Adomavicius und Tuzhilin wie folgt formalisiert [6]:

$$U \times I \mapsto R,$$

wobei R in den meisten Fällen im Intervall $[0 \ldots 1]$ liegt.

In einigen Fällen, wie z.B. dem wissensbasierten Ansatz sind die Werte der Nutzenfunktion a

[1] http://www.amazon.de

priori bekannt, in anderen Fällen wie z.B. dem kollaborativen Ansatz müssen die Nutzenfunktionen aus der Gesamtheit aller Nutzerangaben geschätzt werden [86]. Dabei besteht das Hauptproblem für die Empfehlungsfunktion darin, dass die Nutzenfunktion meist nur auf einem Ausschnitt definiert bzw. für eine Untermenge bestimmt werden kann. Dies liegt in der Tatsache begründet, dass zum einen nicht alle Nutzer eines Empfehlungssystems ihre Präferenzen angeben und andererseits nicht alle Elemente definiert sind, so z.B. für neue oder den Nutzern unbekannte Elemente.

Das Auswahlproblem eines Empfehlungssystems ES kann hingegen als Identifikation der wichtigsten n Elemente aus der Menge aller Elemente I betrachtet werden, wobei für den Nutzer u die höchsten Nutzwerte erzielt werden:

$$ES(u,n) = \{i,\ldots,i_k,\ldots,i_n\} \text{ mit } i_1,\ldots,i_n \in I \text{ und}$$

$$\forall k\ empfehlung\,(u,i_k) > 0 \wedge empfehlung\,(u,i_k) > empfehlung\,(u,i_{k+1})$$

Somit liefert ein Empfehlungssystem eine Ranking-Liste von Elementen, die die höchsten Nutzwerte für einen Nutzer haben. Im Folgenden soll kurz auf die unterschiedlichen Techniken für Empfehlungssysteme eingegangen werden, vergleiche hierzu [40].

Kollaboratives Filtern Das Empfehlungssystem generiert die Empfehlungen aufgrund der Bewertungen unterschiedlicher Nutzer. Bewertungen können dabei explizit in Bewertungsdatenbanken durch den Nutzer oder implizit durch technische Beobachtungssysteme hinterlegt werden. So ist der Kauf eines Buches ebenso eine in diesem Kontext anzusehende Bewertung wie die 4-Sterne Bewertung für das Buch durch einen Leser. Eine persönliche Empfehlung kann gegeben werden, wenn ähnliche Präferenzen in der Datenbasis auftreten. Dieses Verfahren wird beispielsweise im experimentellen Mailsystem Tapestry [70] angewendet. Für einen umfassenden Überlick siehe [134].

Inhaltsbasierter Ansatz In diesem Ansatz werden zwei Quellen genutzt: Einerseits die beschreibenden Merkmale des Produktes und andererseits die Bewertungen dieser durch die Nutzer. Eine Empfehlung wird dabei als Klassifikationsproblem behandelt, wobei der Nutzer durch seine positiven und negativen Bewertungen der Produkteigenschaften die Datenbasis schafft. Als ein Beispiel für den inhaltsbasierten Ansatz präsentieren Jennings und Higuchi ein neuronales Netzwerk, um Nutzerbewertungen unter Unsicherheit besser zu modellieren [88].

Demografischer Ansatz Wird das demografische Nutzerprofil als Ausgangsbasis für das Empfehlungssystem genutzt, spricht man von einem demografischen Empfehlungssystem. Dies lässt sich insbesondere für Nischenprodukte und -märkte einsetzen, in denen die demografischen Eigenschaften als Ersatz für die unzureichende Datenbasis dienen. Daher werden in solchen Fällen die Nutzerbewertungen innerhalb der Nischen kombiniert. Mit „Waldo der Webzauberer“ präsentiert Krulwich einen Vertreter des demografischen Ansatzes, der Nutzern Webseiten basierend auf den Nutzerinformationen empfiehlt [99].

Wissensbasierter Ansatz Für diesen Ansatz sind sowohl die Nutzerbedürfnisse wie auch die Nutzerpräferenzen wichtig. Anhand dieser wird mittels Inferenz eine Produktempfehlung vorgenommen. Dabei kann es neben den häufig implizit modellierten Wissensrepräsentationen auch explizite funktionale Wissensbeschreibungen geben, wie genau ein Produkt die Nutzerbedürfnisse erfüllt. Burke nennt an dieser Stelle zudem den nutzenbasierten Ansatz zwar explizit, in der weiteren Betrachtung wird dieser dann aber gemeinsam mit dem wissensbasierten Ansatz behandelt [40].

Ein fallbasiertes Empfehlungssystem ist „Wasabi, der Einkaufsberater“ [39], der für elektronische Produktkataloge entworfen wurde.

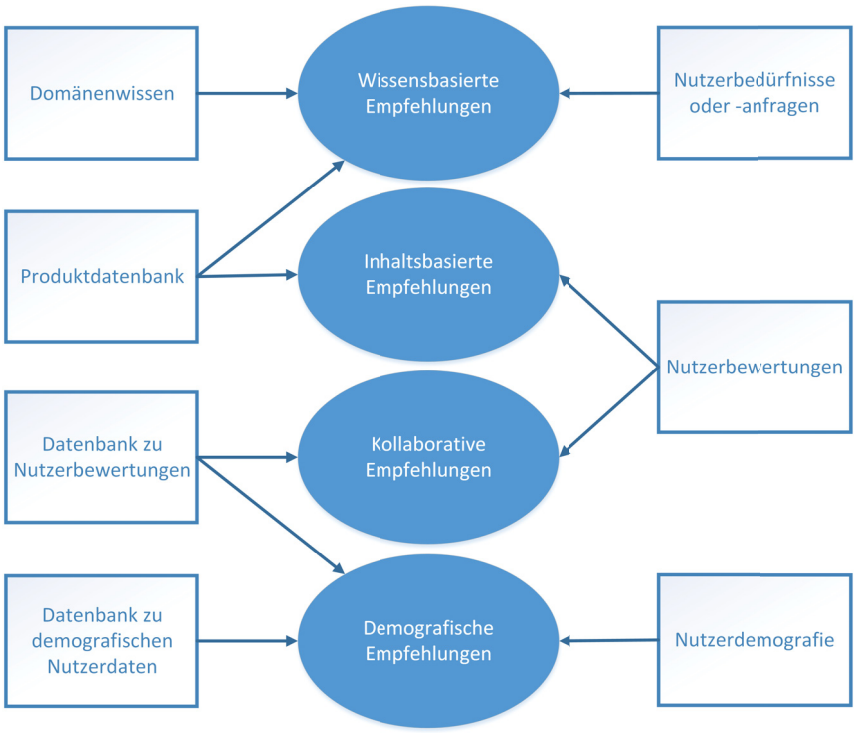

Abbildung 2.1: Techniken für Empfehlungssysteme nach [41]

Abbildung 2.1 verdeutlicht die unterschiedlichen Datenbasen, die für die einzelnen Methoden der Empfehlungssysteme von Bedeutung sind. Hierbei wurde die gleiche Darstellung wie bei Burke verwendet. Während der linke Teil die Datenbasen darstellt, die für das Empfehlungssystem notwendig sind, wird im rechten Teil der Abbildung aufgezeigt, welche Informationen zu einer personalisierten Empfehlung führen.

Hybrider Ansatz Auch die Kombination der unterschiedlichen Ansätze stellt eine Möglichkeit dar, ein Empfehlungssystem zu gestalten. Dabei spielen verschiedene Techniken der Zusammenführung eine Rolle. Burke stellt sieben Möglichkeiten vor, die einzelnen Ansätze zusammenzubringen:

- Bei der *Wichtung* wird der Empfehlungswert entsprechend den festgesetzten Gewichten der Einzelkomponenten ermittelt. [46] stellen für den Anwendungsfall einer Online-Zeitschrift einen Mix aus inhaltsbasiertem und kollaborativem Ansatz vor. Dabei wird insbesondere der Abdeckungsgrad und die Schnelligkeit des inhaltsbasierten Ansatzes mit den Ergebnissen der Kollaboration zusammengebracht.

- Das System trifft eine *Auswahl*, welche Empfehlungskomponente zum Einsatz kommt und wendet diese an. [25] nutzen zwei Agentensysteme, eines für personalisierte Empfehlungen und eines ausgerichtet auf mobile Endgeräte. Sowohl implizites als auch explizites Nutzerfeedback wird genutzt, um das Nutzermodell zu lernen, wobei zwei unterschiedlich ausgerichtete Modelle zum Einsatz kommen, eines für kurzfristige, das andere für langfristige Interessen. Durch das direkte Feedback kann eine Auswahl auch auf akute Nutzerpräferenzwechsel erfolgen.

- Alle Ergebnisse der Einzelkomponenten werden in einer Gesamtdarstellung *gemeinsam* dargestellt. Ein personalisierter TV-Guide wird in [140] vorgestellt, der die Informationsflut der TV Angebote auf Basis der Nutzerpräferenzen filtert. So werden nutzerspezifische TV-Angebote unterbreitet, basierend auf einer inhalts- und kollaborationsbasierten Empfehlung im Mix mit Techniken der Nutzerprofile.
- Einzelne *Eigenschaften* werden *kombiniert* und somit eine neue Basis geschaffen, auf der dann die Empfehlungskomponente angewendet wird. In [17] werden neben den Nutzerbewertungen zusätzlich verfügbare Informationen genutzt, die in Kombination die Nutzerpräferenzen vorhersagen sollen. Als Beispiel dient ein Filmempfehlungsdienst, der mit dem Social-Filtering System Recommender [81] verglichen wird.
- Bei der *Eigenschaftsfortpflanzung* werden zunächst durch eine Komponente die Eigenschaften berechnet, um diese dann als Eingabe für die nächste Komponente zu verwenden. Um die Nachteile der kollaborativen sowie inhaltsbasierten Empfehlungssysteme zu überwinden, werden in [112] mittels inhaltsbasierten Empfehlungen die Nutzerdaten angereichert, um dann personalisierte Vorschläge zu unterbreiten.
- Beim *Stufenverlauf* gibt es eine definierte Reihenfolge der Komponenten, wobei jedoch niedriger priorisierte Komponenten vorher identifizierte Verbindungen aufbrechen können. Die Datenvielfalt stellt Empfehlungssysteme vor eine Herausforderung, welcher mit einer Anpassung bzw. Abschwächung der Ähnlichkeitsmaße entgegengewirkt wird. In [111] wird ein ähnlichkeitserhaltender Ansatz vorgestellt, der bei gegebener Änhlichkeit die Vielfalt maximiert oder etwas Ähnlichkeit reduziert, um die unterstützte Vielfalt weiter zu erhöhen.
- Auf dem *Meta-Level* wird das durch eine Komponente berechnete Modell als Eingabe für die nächste Komponente genutzt. Im Kontext der personalisierten Webseitenempfehlung wird in [125] ein System vorgeschlagen, dass sowohl kollaborativ, inhaltsbasiert und demografisch arbeitet. Am Beispiel der Restaurantempfehlung wird aufgezeigt, dass nur das Ranking der einzelnen Empfehlungsstufen genutzt wird und nicht die Stärke der Empfehlungen.

Bei der Kombination unterschiedlicher Empfehlungssysteme / -komponenten ist die Reihenfolge zu beachten. Daher ergeben sich theoretisch 84 unterschiedliche Möglichkeiten, zwei Verfahren miteinander in einem hybriden System zusammenzubringen. Jedoch sind einige nicht möglich, wie das kollaborative Filtern mit dem demografischen Ansatz mittels Meta-Level zu kombinieren. Zudem sind andere unabhängig der Reihenfolge, wie z.B. die Wichtung von inhaltsbasiertem Ansatz und kollaborativem Filtern. Es fallen somit 31 Möglichkeiten weg, so dass 53 mögliche Zweier-Kombinationen offen bleiben [41].

2.4 Data Mining und Empfehlungssysteme

Empfehlungssysteme dienen dem Zweck der Informationsfilterung. Gerade bei großen Datenmengen ist dies erforderlich. Ein sich allgemein in der Praxis durchgesetztes Vorgehen wird durch den Wissensentdeckungsprozess (Knowledge Discovery Prozess (KDP)) [64] beschrieben. Dieser ist in Abbildung 2.2 dargestellt. Während Ansätze im Kontext der Data Warehouses diesen Prozess unmittelbar in ihre Systemumgebung einbauen [78, 97] kann eine Wissensentdeckung auch ohne ein Data Warehouse erfolgen. Für das Buchausleih-Empfehlungssystem soll kein eigenes Data Warehouse aufgesetzt werden. Daher wird in dieser Arbeit der Wissensentdeckungsprozess analog zu Fayyad et al. präsentiert [64].

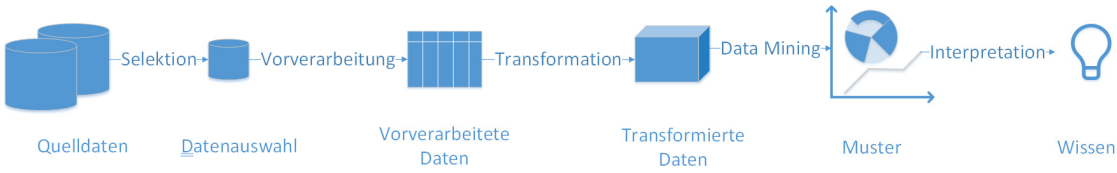

Abbildung 2.2: Der Knowledge Discovery Prozess nach [118]

Die einzelnen Schritte im KDP sind eher sequentiell abzuarbeiten. Jedoch kann es vorkommen, dass aufgrund unzufriedenstellender Ergebnisse eine Anpassung der vorangegangenen Schritte erforderlich ist. Somit kann der KDP auch als sehr iteratives Vorgehen betrachtet werden. In der Abbildung 2.2 wurde der besseren Übersichtlichkeit auf die Darstellung dieser Iterationen verzichtet. Von jedem einzelnen Ergebnis kann auf alle vorangegangenen eine neue Iteration erfolgen.

Der sequentielle Ablauf stellt sich folgendermaßen dar [64]:

1. In der *Selektion* werden die Daten aus allen Quelldaten ausgewählt, die für die weitere Untersuchung verwendet werden sollen. Dies kann als Kontextdefinition angesehen werden.

2. Es schließt sich die *Vorverarbeitung* der Daten an. Dabei handelt es sich im Wesentlichen um die Beseitigung von Qualitätsproblemen in den Daten. Diese können sowohl in der Metadatenebene wie auch der Datenebene auftreten und sind nicht nur durch die Heterogenität der Quelldaten verursacht, sondern können auch durch unzureichend definierte Systemumgebungen auftreten. Aufgrund der Komplexität und Vielfältigkeit wird an dieser Stelle für einen Überblick auf [18, 45, 97] verwiesen. Das Ergebnis sind häufig relational vorliegende Daten.

3. In der *Transformation* werden die Daten für das entsprechend anschließende Verfahren aufbereitet. Dies kann z.B. durch Datentypkonvertierungen, aber auch durch Datenbereichsänderungen erzielt werden. Anschließend liegen die Daten meist in einer multidimensionalen Repräsentation vor, die für die unmittelbare weitere Bearbeitung effizient ist.

4. Die Ausführung des *Data Mining* Algorithmus auf den Daten mit einer entsprechenden Parametrisierung führt zu einem Modell, dass eine Abstraktionsebene darstellt und somit meist Muster hervorbringt, die die Daten näher beschreiben bzw. diese identifizierbar machen.

5. Letztlich muss auch eine Bewertung und *Interpretation* der gewonnenen Muster erfolgen. Hierbei spielen die Untersuchungen von Struktur und Auswirkung eine entscheidende Rolle. Eine aufbereitete Darstellung kann dann in Wissensbasen abgelegt werden.

Bei diesem Vorgehen handelt es sich um eine explorative Analyse, so dass die lineare Abarbeitung in der Praxis häufig nicht zur Anwendung kommt. Für die Entdeckung von Zusammenhängen bedeutet dies einen hohen Aufwand bei der Erstimplementierung. Zugleich bietet das Vorgehen dann aber auch an, in einen linearen Kontext zu wechseln, so dass bei täglicher Änderung des Datenbestandes nur wenig Aufwand notwendig ist, der sich insbesondere auf die Abweichungsanalyse zu den bisherigen entdeckten Mustern ergibt. Somit ist das Anwenden in einem Buchausleih-Empfehlungssystem automatisiert möglich, sobald die Initialisierungsphase abgeschlossen ist. Diese Arbeit beschäftigt sich vornehmlich mit den Fragen der Initialisierung, daher erfolgen unterschiedliche Iterationen insbesondere in Kapitel 5. Bereits seit den 70er Jahren ist im Marketing die Analyse von Produkten, die gemeinsam gekauft werden von Interesse. Einen ersten datenbasierten Ansatz liefert Böcker mit der Nutzung der Verbundintensitäten [28]. Anstelle der reinen singulären Kostenbetrachtungen werden Produkte auch betrachtet, die entweder durch den Nutzerbedarf, eine One-Stop-Shop-Strategie

oder ein absatzpolitisches Instrument gemeinsam gekauft werden. Mit den Verbundkoeffizienten und der Verwendung von Korrelationen und Faktorenanalyse (für einen umfassenden Überblick siehe u.a. [13]) ergeben sich dabei Einsichten im unmittelbaren Produkt- bzw. Sortimentsvergleich.

Hruschka verwendet in der Weiterentwicklung Logit-Modelle [83], so dass ein probabilistischer Ansatz genutzt wird. Dies bedeutet zugleich eine Änderung der Interpretationsebene von Vierfeldertafeln hin zu einer multidimensionalen Datenanalyse.

Der nächste entscheidende Durchbruch wird mittels Anwendung von Data Mining Algorithmen im Kontext der Verbundanalyse erzielt [51]. Insbesondere das Verfahren von Agrawal und Srikant, der Apriori Algorithmus [7], verbessert den Zugang zu Produktinformationen. Dieser Algorithmus wird in Kapitel 3.4 detailliert vorgestellt. Weiterentwicklungen wie die Einbeziehung der Rough-Set-Theorie nach [124] konnten zwar Verbesserungen in der Performanz hervorrufen [155], sind aber in den aktuellen Data Mining Umgebungen selten vertreten.

Assoziationsregeln dienen der Identifikation von Korrelationen zwischen gemeinsam auftretenden Elementen [30]. Dabei sind keine Annahmen über die Zusammenhänge und die einzelnen Dinge (Items) notwendig. Die Stärke der Korrelationen (Zusammenhänge) zwischen den Items wird als Konfidenz bezeichnet. Die Information über die Häufigkeit der Items wird als Support bezeichnet. Die Algorithmen zur Assoziationsanalyse werden daher mit zwei Parametern gestartet, dem Mindestsupport und der minimalen Konfidenz. Diese sind jeweils sehr abhängig von den Datenbeständen und der anschließenden Verwendung der Regelmenge. Daher wird in Kapitel 5 eine Untersuchung diesbezüglich erfolgen.

Aber auch die Verbundanalyse wurde weiterentwickelt, denn auch zur Jahrtausendwende bestand noch Uneinigkeit über die Best-Practices in diesem Gebiet [51]. Mit besseren rechnergestützten Verfahren sind auch neuere und komplexere Analysemethoden möglich. Eine Übersicht zu den Modellen und Visualisierungsmethoden ist in Tabelle 2.1 wiedergegeben.

Tabelle 2.1: Modelle und Visualisierungsmethoden der Verbundanalyse nach [51]

	60er/70er Jahre	80er Jahre	90er/00er Jahre
Modellierung	Kreuzelastizitäten Lineare Programmierung Bedingte Wahrscheinlichkeiten Assoziationskoeffizienten Korrelationskoeffizienten Lineare Regression	Polynomial-Lag-Korrelation Logistische Regression	Multivariate Logitmodelle Neuronale Netze Binomialverteilungsmodell Assoziationsregeln
Visualisierung	Multidimensionale Skalierung Clusteranalyse Faktorenanalyse		Self-organizing maps Hypergraphen

Für die Bestimmungsfaktoren der Verbundanalyse existieren eine Vielzahl von Modellen und Modellannahmen, die übersichtlich in [33] dargestellt sind. Dabei wird sowohl auf Probit- wie auch Logit-Modelle eingegangen. Da in der vorliegenden Arbeit der Fokus insbesondere auf Assoziationsverfahren für die Warenkorbanalyse gelegt wird, (siehe Abschnitt 3.4) werden an dieser Stelle nur der Vollständigkeit halber die anderen Modelle der Verbundanalyse kurz vorgestellt.

Die Self-Organizing Maps (auch Kohonen-Netze genannt) sind eine Visualisierungsmethode, die annimmt, dass Verbundbeziehungen mit ähnlichem Muster auftreten und nur auf einer gemeinsamen Item-Ebene sinnvoll darstellbar sind [51, 52]. Dabei werden ähnliche Muster gemeinsam in einem Cluster zusammengefasst. Dieser Ansatz benötigt als Eingabeparameter die Mindestanzahl an Items, die in den Regeln betrachtet werden sollen. Die grafische Darstellung erfolgt dabei als zweidimensionale Matrix.

Multikategorielle Entscheidungsmodelle sind ein weiterer Ansatz, um das Kaufverhalten zu untersuchen. Im Vordergrund stehen dabei Produktkategorien und ihre Einflüsse aufeinander. Für eine Übersicht möglicher Abhängigkeitsstrukturen der Produktkategorien siehe [131]. Unterschiedliche Ansätze für multikategorielle Entscheidungsmodelle, die an dieser Stelle nur erwähnt werden sollen, können dabei wie folgt unterschieden werden [136]:

- *Kaufentscheidungen* können sowohl von komplementären wie auch substitutiven Gütern abhängen. Für die Produkte in unterschiedlichen Kategorien können unter anderem das multivariate Probit- oder Logit-Modell eingesetzt werden.
- Die Frage nach dem *Zeitpunkt der Kaufentscheidung* kann mittels multivariatem Hazard-Modell gelöst werden. Hierbei steht der Kaufzeitpunkt im Vordergrund und somit eine kontinuierliche Modellierung des Kaufwagnisses als zeitliche Abhängigkeit.
- *Bundle-Auswahl* bedeutet die Kaufentscheidung für einen zusammengesetzten Produktkorb. Hierbei werden die Modelle auf Bundle-Ebene und nicht auf der Kategorienebene erstellt.
- Mit der *Korrelation der Werbungsbefindlichkeiten* wird zwischen den Produktkategorien geprüft, wie abhängig das Markenverhalten der Käufer ist. Ein Modell zur Auflösung der Varianzerklärung kann an dieser Stelle zielführend sein, um sowohl die Haushaltsspezifika wie auch die Produktspezifika zu trennen.
- Die *Korrelation von Produktmarken in unterschiedlichen Kategorien* ist ein weiterer Ansatz der multikategoriellen Entscheidungsmodelle. Sowohl Poisson-Prozesse für das Markenumsatzwachstum wie auch multinominale Logit-Modelle dienen zur Identifikation der Zusammenhänge.
- Ebenfalls ist es möglich, Aussagen über die *Kaufmenge* zu treffen. Dazu bieten sich insbesondere Verfahren an, die auf die Drichlet-Verteilung basieren, die den Anteil am Gesamtwarenkorb modellieren.

Darüber hinaus lassen sich auch Kombinationen zusammensetzen, die z.B. auf Vorhersagen aufgrund der Marken innerhalb der Produktkategorien und der Menge aufbauen. Für einen umfassenderen Einblick wird auf [136] verwiesen.

Die meisten Transaktionen werden durch OLTP-Systeme gespeichert. Daher bietet es sich an, die Datenanalyse direkt in den transaktionalen Kontexten zu beginnen. Problematisch ist an dieser Stelle jedoch die Tatsache, dass die vorliegenden Daten für das Data Mining meist nicht hinreichend gut abgelegt sind. Dies bedeutet dann einen hohen Leseaufwand, der ggf. für den Betrieb der OLTP-Systeme gravierende Folgen aufweisen kann. Daher ist die Trennung von analytischem System, zu dem Data Mining Verfahren zählen, und operativem Geschäftssystem dringend notwendig. Auch in der Umgebung wissenschaftlicher Bibliotheken ist es erforderlich, diese System zu entkoppeln, damit eine zufriedenstellende Leistung gegeben ist. Somit wird in der vorliegenden Arbeit eine Datenentnahme aus dem operativen System erfolgen, so dass dieses weitestgehend unberührt vom zusätzlichen Dienstangebot betrieben werden kann.

2.5 Empfehlungssysteme für wissenschaftliche Bibliotheken

Im folgenden Abschnitt soll kurz auf die in der Praxis und wissenschaftlichen Literatur behandelten Systeme für wissenschaftliche Bibliotheken eingegangen werden. Dabei kann ein vollständiger Abriss nicht erfolgen, sondern es wird versucht, eine möglichst hinreichend abdeckende Auswahl zu

diskutieren. Für einen Überblick zu den Möglichkeiten von Text- und Data Mining Ansätzen in wissenschaftlichen Bibliotheken siehe auch [56].

Gärtner beschreibt die gesamte Spannbreite von Empfehlungssystemen im deutschsprachigen Raum. Dabei werden sowohl kommerziell genutzte wie auch frei verfügbare Systeme dargestellt [68]. Hierbei inkludiert sie den Bereich der Recommendersysteme für Bibliotheken unter anderem mit der am KIT entwickelten Lösung *BibTip*[116], der an der Universität Kassel entwickelten Lösung *Bibsonomy* [121] und der zunächst für den privaten Gebrauch entwickelten Lösung *Library Thing* von Tim Spalding. Während sowohl manuelle Empfehlungssysteme, wie z.B. Bestsellerlisten und Bibliographien angerissen werden, stehen bei Gärtner automatisierte Empfehlungssysteme im Vordergrund. Die unterschiedlichen Empfehlungssysteme klassifiziert sie analog zu Burke und verfeinert diese insbesondere im Kontext webbasierter Lösungen.

Für die Klasse der inhaltsbasierten Empfehlungen, stellt sie beispielsweise RSS-Feeds für Themensuchen im OPAC vor. Die Onleihe[2] bettet sie in die Klasse der demografischen Empfehlungssysteme. Für die weitere Betrachtung sollen insbesondere die Systeme vorgestellt werden, die in wissenschaftlichen Bibliotheken zum Einsatz kommen.

BibTip stellt einen hybriden Ansatz dar, der direkt in den OPAC integriert ist. Hierbei werden die Suchanfragen im Katalogsystem genutzt, um Empfehlungen abzugeben [116]. Während Nutzungsstatistiken der Suchanfragen genutzt werden, spielen Ausleihdaten keine Rolle. Dies führt vor allem dazu, dass auch nicht relevante Bücher empfohlen werden können, die nur bei der Suche identifiziert wurden, aber kein Interesse hervorrufen. Im Kontext der Verwendung des Empfehlungssystems kann dies beim Nutzer dazu führen, dass empfohlene Bücher stärker betrachtet werden, auch wenn sie das eigentliche Interesse nicht widerspiegeln. Dies führt dann dazu, dass aufgrund des Navigationsverhaltens Empfehlungen gemacht werden, die aufgrund des Empfehlungssystems dann verstärkt angeklickt und wiederum zu einem höheren Score in der Empfehlung führen.

Das System setzt dabei auf die von Ehrenberg entwickelte Theorie der wiederholten Käufe [60]. Als Vorteile gegenüber einem Empfehlungssystem, das direkt an das Ausleihsystem gekoppelt ist, werden insbesondere der Datenzugang und die wesentlich größere Zahl der betrachteten Treffer im OPAC angegeben [117]. Jedoch ist es mit der Novellierung des EU-Datenschutzrechts zumindest fragwürdig, ob die Verwendung der IP-Adressen für die Nutzeridentifikation zulässig ist. Auch die geringere Wahrnehmung der Nutzer hinsichtlich der Zeit aufgrund der Sessionbildung wird in der vorliegenden Arbeit untersucht. Dabei wird die Vermutung überprüft, dass kürzere Sessions zu mehr Warenkörben führen, dies jedoch mit dem Verlust der langfristigen Zusammenhänge und somit einer geminderten Empfehlungsqualität einhergeht.

Es ist auch möglich, dass das Katalogsystem, insbesondere durch den verstärkten Einsatz von Discovery-Systemen, zu einer Reduzierung der Nutzerbasis und somit zu einer geringeren Nutzung und Repräsentativität führt. Daher sollte ein Empfehlungssystem diese unterschiedlichen Suchsysteme gemeinsam abdecken. Aufgrund der Einbindung in Metakataloge, die eindeutige Identifizierer wie die Pica Produktionsnummer (PPN) verwenden, ist es möglich, eine erhöhte Anzahl an Empfehlungen zu erstellen. Dies führt durch eine Ablösung von individualisierten Angeboten der jeweiligen wissenschaftlichen Bibliothek hin zu einer globalen Empfehlung. Dieses Angebot inkludiert dann beispielsweise auch Fernleiheempfehlungen.

Bibsonomy setzt sich aus den Begriffen Bibliographie und Folksonomy zusammen. Dieser Ansatz kann als Social-Bookmark-Dienst betrachtet werden [68], der sich insbesondere den wissenschaftlichen Publikationen widmet. Die Kernidee basiert dabei anstelle der Dreier-Tupel von Empfehlungssystemen auf einem Fünfer-Tupel aus dem Kontext der Folksonomies, basierend auf: Nutzern, Tags und

[2] http://www.onleihe.net/

Dingen sowie der Beziehung zwischen diesen drei Bestandteilen und einer nutzerspezifizierten Taghierarchie [121]. Unter Verwendung der Tags ergeben sich dann weitere Beziehungszusammenhänge und Filtermöglichkeiten.

Auch das kommerzielle Tool *Library Thing* setzt auf den Tagging Ansatz. Hierbei werden eigene oder öffentliche Publikationssammlungen durch Nutzer getaggt. Das Grundkonzept ist der Gedanke, dass Klassifizierungssysteme wie der Library of Congress (LoC) oder die Regensburger Verbundklassifikation (RVK) ungeeignet sind, private Sammlungen zu ordnen. Daher werden beliebige Tags für die Beschreibung der Publikationen genutzt. Durch die Kommerzialisierung ergeben sich aber auch Nachteile, die beispielsweise Neumann diskutiert [121]. Insbesondere der Mismatch der Personengruppen, hinsichtlich der Tagger und der einfachen Nutzer, sowie die extreme Vielfältigkeit der Tags im Vergleich der wissenschaftlichen Klassifikationssysteme werden bemängelt. Der Nutzen von Library Thing sowohl für die Werbung von Büchern wie auch in wissenschaftlichen Bibliotheken wird in [128] untersucht. Dabei ist das System nicht geeignet, um Nutzer an die Bibliothek bzw. das System zu binden, sondern vielmehr können Buchpakete gezielt darüber vermarktet werden. Der Einsatz von Library Thing im Webkatalog wird unter Berücksichtigung der Nutzungsstatistiken in [113] betrachtet.

Eine Untersuchung zu 329 OPAC-Systemen in Großbritannien wird in [152] vorgestellt. Dabei wird deutlich, dass die Mehrzahl der Kataloge keine Empfehlungssysteme nutzen. Nur etwa 11% der wissenschaftlichen Bibliotheken in der Untersuchungen bieten Empfehlungen im OPAC an. Dabei werden direkt BibTip und Library Thing näher beleuchtet. Die Studie gibt die Empfehlung, dass Bibliotheken den Nutzerfokus stärken und somit den Einsatz von Empfehlungssystemen intensivieren sollen.

In [44] werden Katalogdienste untersucht, die es dem Nutzer ermöglichen, Tagging sowie Bewertungen und Rezensionen innerhalb der Bibliothekskataloge (z.B. OPAC) oder auf eingebetteten Webdiensten, wie Library Thing, abzugeben. Als wichtigstes Ergebnis ist dabei anzumerken, dass die Informations- und Systemqualität an oberster Stelle für den Erfolg bei den Nutzern stehen. Für die kontinuierliche Nutzung (sowohl aktiv als auch passiv) sind dabei der Zusammenhang zwischen der Informationssystem-Erfolgstheorie (Information System Success Theory) nach DeLone und McLean [53], dem Gemeinschaftsgefühl [110] und der Theorie der „geplanten“ weiteren Nutzung von Katalogwebseiten [160] relevant.

Die von 70000 Nutzern bei der Ausleihe von einer halben Million zur Verfügung stehenden Büchern produzierten 5 Millionen Datensätze dienen [66] als Basis, um ein nutzerbasiertes und kollaboratives Empfehlungssystem für die Ausleihe zu entwickeln. Dies wird damit begründet, dass fachspezifische Literatur in der gleichen gemeinsamen Domäne ausgeliehen wird. Problematisch ist der recht hohe Zeitraum mit einem Abdeckungsgrad von 14 Jahren.

Mr. DLib ist ein Publikationsempfehlungsdienst für wissenschaftliche Aufsätze [3]. Mr. DLib (Machine-readable Digital Library) ist ein Webdienst der Zugang zu Volltexten und den Metadaten bietet. Dabei werden Webcrawler genutzt, um die Datenbasis zu erzeugen. Die Struktur der Datenbasis baut dabei auf Dokumenten, Personen, Konferenzen und Zeitschriften sowie Organisationen auf [21]. Aufgrund der Verlinkung innerhalb der Metadaten können Empfehlungen abgeleitet werden. Die Implementierung von Mr. DLib in dem Literaturverwaltungsprogramm JabRef [4] wird in [65] beschrieben.

Eine Untersuchung zu Empfehlungssystemen für wissenschaftliche Artikel wird in [22] vorgestellt. Ergebnisse sind unter anderem, dass etwa 55% der Empfehlungssysteme einem inhaltsbasierten

[3] http://mr-dlib.org/
[4] http://www.jabref.org/

Ansatz folgen. 18% sind in die Gruppe der kollaborativen Systeme einzuordnen. Die anderen Empfehlungssysteme werden insbesondere den graphbasierten Ansätzen zugeschrieben. Die sehr unterschiedlichen Ansätze für die Empfehlung von wissenschaftlichen Publikationen von [22] sind für den Kontext der vorliegenden Arbeit eher ungeeignet und werden daher nicht weiter betrachtet.

[20] evaluieren Empfehlungssysteme für wissenschaftliche Publikationen im Kontext der Offline- und Online-Bewertung sowie mit Nutzerstudien. Hierbei treten ggf. Widersprüche auf, die aufgelöst werden müssen, um eine hinreichend gute Bewertung der Empfehlungssysteme zu erhalten. Dabei kommen Beel und Langer zu dem Schluss, dass Offline-Bewertungen im Kontext der wissenschaftlichen Publikationsempfehlung nicht zielführend sind [20]. Vielmehr bieten sich für den Kontext Online-Bewertungen hinsichtlich der Klick-, Verlinkungs- und Zitationszahlen an.

Auch in [149] stehen wissenschaftliche Publikationen im Vordergrund der Empfehlungen. Dabei wird sowohl auf Informationen zur Nutzung wie auch der Zitation zurückgegriffen. Für das stets auftretende Problem der dünnbesetzten Daten für kollaborative Empfehlungssysteme bieten Nutzungsdaten ein etwas besseres Ergebnis. Hingegen liefern zitationsbasierte Daten einen besseren semantischen Kontext in den Empfehlungen. Da beide Datenansätze komplementär erscheinen, ist das gemeinsame Angebot vielversprechend, um eine Vielzahl von Empfehlungen zu erzeugen [150].

[29] evaluieren ein anonymes Empfehlungssystem für die Virtuelle Fakultät der Wiener Wirtschaftsuniversität. Hierbei wird mit der Repeat-Buying Theorie nach Ehrenberg gearbeitet und ein anonymer Empfehlungsdienst entwickelt, der Informationsbundles empfiehlt. Dabei werden Assoziationsverfahren verwendet und sich auf die Kauf- / Besuchsvariable beschränkt, da diese der wohl wichtigste Faktor in der Kaufentscheidung für Produkte ist [60]. Die Datenbasis sind dabei unter anderem Transaktionslogs, die durch Agenten gesammelt und dann zusammengeführt werden [69]. Mittels Agenten wird dann zudem eine Empfehlung ausgesprochen, wobei die Nutzeroberfläche ebenso wichtig ist wie das Antwortzeitverhalten, dass sich mit steigender Datenbasis erhöht.

Ein im Kontext von Data Warehousing und sozialer Netzwerktheorie situiertes Angebot wird in [102] untersucht. Dabei dient OLAP nicht nur der Identifikation von Zusammenhängen, sondern auch dem Browsen und Beobachten von bibliometrischen Informationen. Es werden unterschiedliche Vorgehen untersucht und Ansätze vorgestellt. Aufgrund der Komplexität der Daten bibliometrischer Netzwerke sind Empfehlungen an dieser Stelle jedoch immer mit einer Nutzerintention verbunden und führen zu individualisierten Lösungen, die für die vorliegende Arbeit ungeeignet sind.

Der Einsatz von Linked Open Data (LOD) für Empfehlungssysteme stellt eine Anreicherung dar, die dem Nutzungsverhalten im OPAC zuträglich ist. In [153] werden hierfür neue Empfehlungsstrategien vorgestellt, die insbesondere für wissenschaftliche Publikationen geeignet sind. Für die Nutzerstudie wurden im Fachgebiet Wirtschaftswissenschaft Untersuchungen vorgenommen, wobei leider offenbleibt, wie eine Skalierung insbesondere hinsichtlich der Problematiken der Heterogenität (z.B. Homonyme und Synonyme sowie der Kontext) erfolgen kann. Auch der Einsatz von RDF ist hier wegen der Skalierung noch näher zu betrachten.

In [24] wird ein Ansatz gewählt, das globale System der Gesamtdatenbasis zu verlassen, um für einzelne Untermengen der Datenbasis bessere Empfehlungen zu geben. Dabei gehen die Autoren davon aus, dass in den Daten häufig unterschiedliche Cluster vorhanden sind und eine globale Optimierungsstrategie daher fehlschlagen muss. Insbesondere können dabei Empfehlungen für Nischenangebote und Neuerscheinungen gegeben werden.

2.6 Empfehlungssysteme unter Betrachtung von Data Privacy

Da personalisierte Empfehlungssysteme eine große Datenbasis über den jeweiligen Nutzer benötigen, um wertvolle und exakte Empfehlungen zu erstellen, steht die Offenlegung der Nutzerprofile im strengen Gegensatz zu Data Privacy [115]. Es existiert ein Zusammenhang zwischen der Offenlegung und dem Nutzen des Empfehlungssystems, der auch als Privacy-Personalization-Tradeoff [42] bezeichnet wird. Obwohl der Nutzen der Personalisierung unbestreitbar für die Informationsfilterung ist [129], gehen damit auch rechtliche Aspekte einher [151]. Aufgrund der Komplexität werden in dieser Arbeit personalisierte Lösungen nicht betrachtet, sondern eine anonym nutzbare Lösung angestrebt.

Auch die Betrachtung von Privacy in Empfehlungssystemen ist bereits durch eine Vielzahl von Forschungsarbeiten erfolgt. Empfehlungssysteme bieten zwar Steuerungsmechanismen für die personenbezogenen Daten, dabei bleibt aber offen, ob diese auch die gewünschten Resultate erzielen. In [157] wird aufgezeigt, dass implizit zur Verfügung stehende Nutzerdaten, z.B. Daten zur Ausleihe, mit den Steuerungsmechanismen gute Ergebnisse hinsichtlich der Data Privacy erzielen, jedoch greifen die Mechanismen bei expliziten Nutzerdaten wie Produktbewertungen der Nutzer nicht. Daher werden in der vorliegenden Arbeit nur implizite Nutzerdaten verwendet.

In [87] werden die einzelnen Belange der Data Privacy mit den für Empfehlungssysteme genutzten Informationen zusammengebracht. Die Tabelle 2.2 stellt die Zusammenhänge zwischen dem Belang und der einzelnen Information dar, wobei + einen hohen Einfluss darstellt, o einen mittleren und - einen kleinen Einfluss.

Tabelle 2.2: Informationen für Empfehlungssysteme und Data Privacy Belange nach [87]

Information / Data Privacy Belang	Verhalten	Kontext	Domänenwissen	Metadaten der Items	Verlauf	Empfehlung	Feedback	Sozial	Nutzereigenschaften	Nutzerpräferenz
Datensammlung	+	o	-	-	+	o	+	o	o	o
Datenspeicherung	o	o	-	-	o	o	o	o	o	o
Datenweitergabe / -verkauf	+	+	-	-	+	+	+	+	+	+
Mitarbeiterzugriff	+	+	-	-	+	+	+	+	+	+
Empfehlung	-	-	-	-	+	-	o	o	-	+
Geteiltes Gerät	o	-	-	-	+	+	o	-	-	+
Sicht von außen	-	o	-	-	+	o	o	+	+	+

Anzumerken ist zunächst, dass sowohl Metadaten der Items wie auch Domänenwissen keinen Einfluss auf die Data Privacy haben, da sie prinzipiell zur Verfügung stehen. Die einzelnen Belange können sowohl bei der Datenerhebung als auch in Bezug auf die Dauer der Datenspeicherung von Relevanz sein. Auch die Datenweitergabe oder der Verkauf der Daten an Dritte stellt einen wichtigen Belang dar. Im Empfehlungssystem kann der Zugriff durch Mitarbeiter erfolgen, die somit Zugang zu den persönlichen Daten erhalten. Auch die personalisierten Empfehlungen stellen selbst einen Belang dar, da sie durch die Verwendung der persönlichen Informationen erzeugt werden. Der

Zugang zu den Empfehlungen erfolgt mittels Geräten und im Kontext der Nutzung durch andere ergibt sich ebenfalls ein Belang. Letztlich ist es auch möglich, dass Informationen von außen freier ersichtlich sind als gedacht.

Kryptografische Ansätze bieten sich an, um Data Privacy Anforderungen auch im Kontext des Data Minings einzuhalten. Hierfür müssen jedoch Änderungen an den Datenbasen [8] oder in den Algorithmen erfolgen [135], so dass häufig ein Informationsverlust auftritt oder Beziehungen und Empfehlungen erst mit einem größeren Datenbestand verfügbar sind. Daher werden häufig geringere Parameteranforderungen für Data Mining in Privacy-erhaltenden Umgebungen notwendig sein, um etwa gleiche Regelbestände zu erhalten.

Die Vermeidung von Brüchen in der Datenverarbeitung kann dazu genutzt werden, um auch die Data Privacy Anforderung zu schützen. Für den Fall, dass Daten von einem System in ein anderes transferiert werden können oder müssen, sollte sichergestellt werden, dass die Daten verschlüsselt sind [14]. Dabei kann es ebenso hilfreich sein, Empfehlungen zu erzeugen, die die eigentliche Nutzerinformation nicht preisgeben. Ein homomorpher Verschlüsselungsansatz bei gleichzeitiger Erzeugung von Empfehlungen für kollaborative Nutzerdaten wird in [14] vorgestellt. Die Grundidee ist dabei, dass Empfehlungen auch auf den verschlüsselten Daten erfolgen, da die homomorphe Verschlüsselung die Struktur der Daten beibehält. Die Verschlüsselung kann sowohl auf den Produktinformationen oder den Nutzerinformationen erfolgen [15].

Für kollaborative Empfehlungssysteme, die Empfehlungen nach dem Nächste-Nachbarn-Prinzip generieren, ergeben sich drei Data-Privacy-erhaltende Möglichkeiten [103]:

1. durch Verrauschen der Nutzerdaten im Sinne der Differentiellen Privacy,
2. durch Randomisierung der Ergebnisse in der Nächste-Nachbarn-Suche oder
3. durch Anwendung beider Verfahren.

Gleichzeitig ist es jedoch möglich, dass die Genauigkeit der Empfehlung sinkt. Daher schlagen Lu und Shen ein partitionierendes Verfahren vor, aus dem die Zufallsergebnisse gewählt werden [103]. Damit erhöht sich die Genauigkeit, da die Struktur der Daten besser berücksichtigt wird.

Die Offenlegung von persönlichen Informationen wird häufig nur eindimensional in Form der Summierung oder als Verhältniszahl angegeben. In [94] wird dargestellt, dass es sich eigentlich um einen mehrdimensionalen Kontext handelt. Dabei wird zugleich deutlich, dass Personen in Bezug auf ihre Data Privacy und der Offenlegung persönlicher Informationen in unterschiedlichen Kategorien einzuordnen sind. Unter Einbezug dieser Erkenntnisse können Empfehlungssysteme etabliert werden, die den jeweiligen Nutzerkontext berücksichtigen und somit die entsprechende Offenlegung adaptiv nutzen. Dies führt dann zu Kontext-sensibilisierten Empfehlungssystemen [93]. Die Nutzerfahrungen wie auch -systembewertungen richten sich dabei maßgeblich an den durch das System bereitgestellten Informationen aus. Insbesondere Informationen zur Offenlegung der Nutzerdaten führen aber eher zu Vertrauens- und Zufriedenheitsverlusten, obwohl diese Informationen zugleich als wichtig angesehen werden. Ein anderer Ansatz ist das Berechnen der Assoziationsregeln sowohl in einem lokalen System, z.B. dem OPAC einer Einrichtung, wie auch die Zusammenführung in den globalen Kontext, z.B. in den GVK. Die Fragen hinsichtlich einer verteilten Data Mining Strategie richten sich hierbei auch an die Data Privacy [91], denn das zur Verfügung stellen der lokal erhobenen Daten für den globalen Kontext ist ggf. nicht möglich. Ein weiterer Ansatz wird in [143] vorgestellt, bei dem der Einsatz von drei Subsystemen zu einem Multi-Agenten-System führt. So können sich zwei Systeme der Privacy und dem Risiko widmen, das dritte verwaltet die Daten. Die Kombination der drei Systeme ergibt dann eine Empfehlung basierend auf den Zwischenergebnissen der Subsysteme.

3 Methodischer Ansatz

Während im vorangegangenen Kapitel die wichtigsten Punkte im Kontext der Empfehlung für die Buchausleihe dargestellt wurden, soll in diesem Kapitel der benötigte Werkzeugkasten aufgezeigt werden. Somit erfolgt eine Einschränkung auf die wichtigsten Elemente, die für das folgende Kapitel benötigt werden. Eine Abgrenzung zur Literatur bzw. zu möglichen weiteren Ansätzen wird ebenfalls kurz dargestellt.

Mit den Bestimmungen für Data Privacy auf der einen Seite und einem Empfehlungssystem für die Buchausleihe auf der anderen Seite müssen Entscheidungen sowohl auf konzeptioneller Basis als auch in der Verarbeitung der Daten und dem entsprechenden Empfehlungssystem erfolgen. Dieses Kapitel beschreibt die Details der konzeptionellen Ebene des Empfehlungssystems sowie die Entscheidungen hinsichtlich der Voraussetzungen für die Umsetzung (siehe Kapitel 4).

Zunächst wird die Ausgangslage kurz beschrieben, wobei insbesondere auf die Implementierungsumgebung für das Empfehlungssystem eingegangen wird. Daran anknüpfend werden Fragen der Datenaufbereitung und schließlich der Kontext Data Privacy mittels Anonymisierungsverfahren vorgestellt. Auch die methodische Gestaltung des Empfehlungssystems für die Warenkorbanalyse ist Bestandteil dieses Kapitels.

3.1 Ausgangslage

Eine Umsetzung des Empfehlungssystems soll an der Universitätsbibliothek Magdeburg exemplarisch durchgeführt werden. Dabei wird auf Komponenten zurückgegriffen, die auch an anderen Einrichtungen im Gemeinsamen Bibliotheksverbund (GBV) genutzt werden. Somit ist eine Überführung der Lösung in andere Systemumgebungen im GBV möglich. Die Ausgangslage für das Empfehlungssystem stellt das LBS dar. Dabei ist im Wesentlichen das Ausleihsystem, in dem die Datenhaltung der Nutzerdaten zur Ausleihe enthalten sind, sowie die Darstellung im Katalogsystem OPAC für das Empfehlungssystem relevant. Trotzdem wird die in Abbildung 3.1 dargestellte Systemstruktur auch in anderen Belangen teilweise für die Lösung genutzt. Aufgrund der Komplexität wird an dieser Stelle aber nicht auf alle weiteren Teile eingegangen.

Prinzipiell müssen unterschiedliche Ansätze genutzt werden, um die Anforderungen einer integrierten Lösung unter gleichzeitiger Beachtung der Datensparsamkeit zu realisieren. So kann die Datenentnahme direkt am Datenbanksystem LBSDB erfolgen. Hierbei kann SQL genutzt und die Ergebnisse direkt in eine Datei geschrieben werden. So kann in zyklisch sinnvollen Abständen die für die Warenkorbanalyse notwendige Ermittlung der Buchausleihen erfolgen. Diese müssen in den Datenbestand der Warenkörbe so integriert werden, dass sie in jedem entsprechenden Warenkorb als neu identifizierte Elemente hinzugefügt werden. Diese Duplikateliminierung ist notwendig, da Nutzer ein Buch zwar mehrmals ausleihen können, die effiziente Berechnung der Regeln aber echte Mengen und keine Multimengen erfordert.

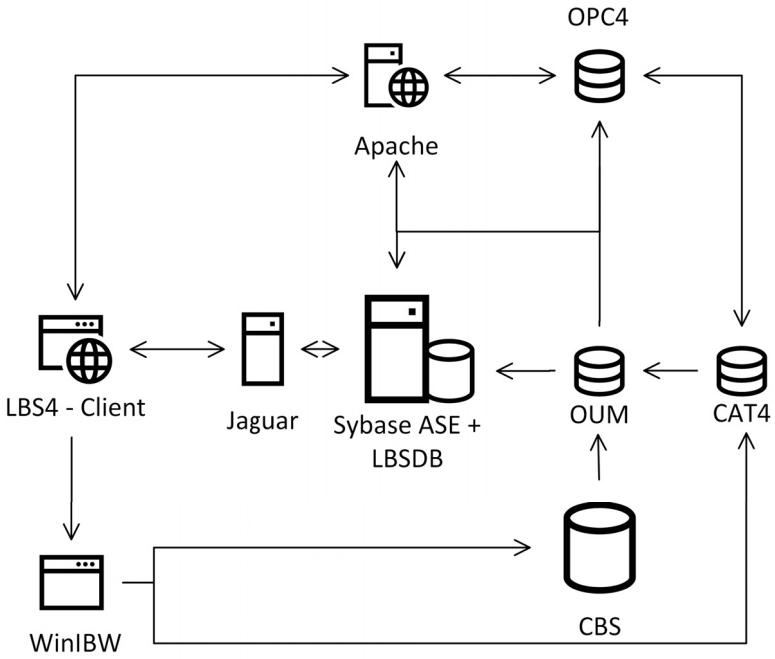

Abbildung 3.1: Systemarchitektur LBS nach [89]

3.2 Datenaufbereitung

Die Datenhaltung im Ausleihsystem OUS erfolgt in unterschiedlichen Tabellen und Sichten. Aufgrund fehlender Metadaten im Datenbanksystem muss zunächst eine Analyse relevanter Daten erfolgen. Um die Gewährleistung der Privacy-Anforderungen zu sichern, erfolgt eine Entkopplung von Ausleihsystem und Empfehlungssystem. Darüber hinaus soll eine eindeutige und nicht umkehrbare Funktion genutzt werden, um einerseits eine Zuordnung aller über die Zeit anfallenden Daten korrekt zu ermöglichen und andererseits unmittelbare personenbezogene Rückschlüsse zu vermeiden. Die Konzeption des Ausleihsystems sieht vor, dass Daten nur für den Zweck der Ausleihe gespeichert werden. Daher handelt es sich dabei um ein klassisches OLTP System [133]. Für ein Empfehlungssystem werden aber insbesondere auch historische Daten benötigt, so dass idealerweise die Nutzung eines Data Warehouse System [97] in Frage kommt. Dies ist auch im Knowledge Discovery Prozess [78] explizit vorgesehen. Jedoch wird an dieser Stelle dieser Aufwand nicht gemacht, da nur ein sehr kleiner Datenausschnitt benötigt wird. Somit werden zur Überführung der Daten einzelne Extract-Transform-Load (ETL) Schritte nach dem Pattern-Ansatz [96] genutzt und an dieser Stelle kurz beschrieben.

Für die Warenkorbanalyse ist es wichtig, Daten über einen Warenkorb zu haben. Hierbei gibt es prinzipiell im Sinne der Empfehlungssysteme zwei Ausprägungen. Einerseits kann jeder Einkauf als abgeschlossene Einheit betrachtet werden. In diesem Fall würden nur Muster gesucht werden, bei denen Items sich gleichzeitig in einem Warenkorb befanden. Dies ist für Szenarien der Filialanalyse oder auch des Cross-Marketings sinnvoll. Für Produkte, die aufgrund ihrer Eigenschaften nur selten gekauft werden, wie z.B. Luxus- oder Investitionsgüter bietet sich dieser Ansatz jedoch nicht an.

Alternativ können die Warenkörbe über einen längeren Zeitraum aggregiert werden. Dies kann sogar so weit gefasst werden, dass der Warenkorb sich über die Zeit immer weiter füllt und die Transaktionsdaten als Streaming Data angesehen werden können. Für das Buchausleihsystem ist der zweite Ansatz zu berücksichtigen. Aber auch das Aufteilen in Zeitintervalle, z.B. auf Jahresscheiben

oder über ein Semester kann sinnvoll sein. Zudem ist es bei dieser Strategie auch möglich zu definieren, ob Intervalle einen Überschneidungszeitraum aufweisen sollen.

Vergessen von Informationen ist im Umfeld der Recommendersysteme ungewohnt, da man keinen Informationsverlust erleiden möchte. Erste Arbeiten in diesem Zusammenhang zeigen jedoch unterschiedliche Möglichkeiten, mit obsoleten Daten umzugehen [109]. Dabei werden ältere Daten nicht unbedingt als obsolet betrachtet. Durch das Vergessen obsoleter Information wird die Vorhersagekraft des Empfehlungssystems signifikant verbessert.

Unterschiedliche Ansätze für das Vergessen dienen dabei als Strategie, wobei eine Unterteilung in zwei Kategorien für die folgenden fünf Strategien möglich ist:

- *Sensitivitätsbasiertes Vergessen*: Hierbei wird untersucht, ob sich ein einzelnes Element der Nutzerbewertung (z.B. eine Buchausleihe durch einen Nutzer) mit den bisherigen konsistent darstellt. Dabei wird unter anderem der Gedanke zugrunde gelegt, dass ein Nutzer auch für andere Nutzer ein Buch ausleiht und diese Ausleihe nicht berücksichtigt werden soll. Während dies auf Nutzerebene erfolgen kann, ist es auch möglich dies im Gesamtempfehlungskontext zu betrachten.
- *Letzte N im Gedächtnis:* Ein sich bewegendes Fenster wird über die Präferenzbasis geschoben und entweder über die Nutzerdatenbasis wie auch die Gesamtdatenbasis bewegt [108].
- *Änderungsidentifikation*: Eine Annahme in Empfehlungssystemen ist, dass sich Nutzerpräferenzen nur langsam ändern. Um abrupte Wechsel ebenfalls adäquat zu berücksichtigen, müssen diese identifiziert und anschließend die vorangegangenen Präferenzen vergessen werden. Während die Messung sowohl auf dem Recall erfolgen kann, ist es auch möglich eine Sensitivitätsanalyse der Regeln durchzuführen, um Ausreißer zu identifizieren.
- *Vergessen (un)populärer Items*: Um weniger wichtige Items aus der Datenbasis zu entfernen, wird eine lineare Transformation angewendet, die weniger wichtigen Items eine geringere Gewichtung im Berechnungsmodell zuweist. Somit landen diese im Empfehlungssystem an nachgeordneter Stelle. Für populäre Items kann die gleiche Strategie angewendet werden.
- *Nutzerfaktor ausfaden*: In dieser Strategie werden die Nutzerpräferenzen in ihrem zeitlichen Kontext bewertet, so dass ältere Präferenzen weniger stark im Präferenzmodell berücksichtigt werden. Ein komplettes Vergessen wird jedoch nicht zugelassen. Während konstante Faktoren den zeitlichen Kontext voll abbilden, können Schwankungen der Nutzereinflüsse auch analog zu den Strategien der Sensitivitätsanalyse oder basierend auf dem Recall verwendet werden, um vor allem Änderungen der Nutzerpräferenzen zu berücksichtigen. Diese Strategieklasse wird auch im Stream Mining [43, 159] verwendet.

Während die ersten drei Strategien sich auf die Datenbasis fokussieren und somit als ratingbasierte Strategien bezeichnet werden können, ändert die zweite Gruppe das Präferenzmodell und kann daher als Latente Faktoren Strategie angesehen werden.

In dieser Arbeit soll ein Vergessen auf Basis älterer Transaktionen untersucht werden, angelehnt an die Fadingstrategie angesehen werden, jedoch wird in dieser Arbeit ein komplettes Wissen zu einem Zeitabschnitt Vergessen oder in der Basis behalten.

3.3 Datenanonymisierung

Datensparsamkeit ist ein wichtiger Grundsatz des Datenschutzes. Das Konzept kann derart umgesetzt werden, dass nur die personenbezogenen Daten gespeichert und verarbeitet werden, die für die

eigentliche Aufgabe essenziell sind. Paragraph 3a des Bundesdatenschutzgesetzes [3] schreibt sowohl Datenvermeidung wie auch -sparsamkeit fest.

Für die Warenkorbanalyse sind die benötigten Eingabedaten die Transaktionen und ihre Items. Aufgrund der in Abschnitt 2.2.2 beschriebenen Datenrückschlüsse muss darauf geachtet werden, dass die Daten nicht so genutzt werden, dass ein Personenbezug herstellbar ist. Daher sollten die Transaktionen mit ihren einzelnen Items entweder so abgelegt werden, dass eine Zuordnung der Personen nicht möglich ist, oder aber aufgrund der Ergebnisrepräsentation dieser Zusammenhang nicht deterministisch gezogen werden kann.

Es ergibt sich die Aufgabe, an zwei Stellen Data Privacy Konzepte zu etablieren. Auf der einen Seite soll die k-Anonymität nach [145] berücksichtigt werden, um die Datenbasis hinreichend abzusichern. Realisiert wird in Kapitel 4 dieser Ansatz durch die Festlegung des Supports. Während der Support von 1 jeden Nutzer differenzieren würde, ergibt sich bei einem Supportwert von 2 bereits die k-Anonymität von 2 und spezifische Nutzer lassen sich nicht mehr unmittelbar unterscheiden. Auf der anderen Seite soll auch im Kontext der Anfrageergebnisse Data Privacy berücksichtigt werden. Hierzu wird im Sinne der Differential Privacy [57] eine gefilterte Ausgabe auf zwei Stufen realisiert. Es wird einerseits nur ein definierter Anteil der ermittelten Regeln in die Datenbasis gespielt. Zudem wird eine randomisierte Teilausgabe dieser präsentiert. Durch diese beiden Maßnahmen wird die Anfrage nicht vollumfänglich bedient, jedoch ausreichend, denn der Nutzer fordert nicht alle Empfehlungen. Dies ist analog der Google-Suchergebnisse zu sehen, die auch eine randomisierte Ausgabe der Ergebnisse präsentieren [23].

Kryptologische Hashfunktionen

Eine Möglichkeit Daten zu anonymisieren, ist die Verwendung von eindeutigen, nicht-umkehrbaren Funktionen. Eine in der Kryptologie und den Prüfverfahren häufig zum Einsatz kommende Familie sind die Secure Hash Algorithm (SHA). Sie werden genutzt, um Prüfwerte für digitale Daten zu erstellen. Die wichtige und für die Warenkorbanalyse nutzbare Eigenschaft der Kollisionssicherheit bedeutet, dass es unwahrscheinlich ist, dass zwei gehashte Daten denselben Prüfwert erhalten. Somit wäre die Eindeutigkeit gewährleistet. Zugleich ist eine Rücktransformation der Hashwerte auf die Ursprungswerte nicht möglich. Dies garantiert im Sinne der Anonymisierung ein hohes Maß, sodass Rückschlüsse auf personenbezogene Kontexte nicht möglich sind, da keine identifizierenden Merkmale mehr existieren. Somit kann davon ausgegangen werden, dass der Personenbezug der Daten nicht mehr vorhanden ist.

Mit der technischen Entwicklung der Systemlandschaften haben sich über die Zeit Veränderungen ergeben. Diese müssen im Kontext der Datenanonymisierung berücksichtigt werden. So waren Hashverfahren wie der Message-Digest Algorithmus 5 (MD5) von Rivest [130] bereits nach zwei Jahren als nicht mehr kollisionsfrei identifiziert [54]. Auch im Bereich der SHA-Familie sind die Entwicklungen soweit vorangegangen, dass momentan der SHA-256 als sicher angesehen werden kann. Dieser wird in der vorliegenden Arbeit genutzt und daher vorgestellt. Jedoch sind für zukünftige Entwicklungen auch andere Hashverfahren einsetzbar und können bedarfsweise genutzt werden.

Der SHA-256 Algorithmus funktioniert ähnlich den Vorgängern, MD5 und SHA-1. In Algorithmus 1 wird das Verfahren nach Eastlake und Hansen vorgestellt [59].

SHA-256 funktioniert zweistufig. Zunächst wird die Nachricht N so aufgefüllt, dass das Ergebnis ein Vielfaches von 512 Bits darstellt und dieses in n 512 Bit-Blöcken der Form $N^{(i)}$ abgelegt wird (Zeilen 1 - 8). Die Notation $N_0^{(i)}$ bezeichnet dabei die ersten 32 Bit von $N^{(i)}$ und $N_{15}^{(i)}$ die letzten 32 Bit. Somit kann mit Operationen auf 32 Bit-Wörtern gearbeitet werden.

An die Datenvorbereitung schließt sich die eigentliche Hash-Berechnung an. Hierzu werden sowohl

Input : Nachricht N
Result : SHA-256 Nachricht

$l \leftarrow$ Länge von N in Bit
$k \leftarrow k > 0 \wedge min_k\{l + 1 + k\} \equiv 448 \; mod \; 512$
resultat $\leftarrow N + 1$
for $i \leftarrow 1$ **to** k **do**
  resultat $\leftarrow$ resultat $+ 0$
end
resultat $\leftarrow$ resultat $+ [l$ in 64 Bit$]$
Zerlege resultat in n 512 Bitblöcke $N^{(1)}, \ldots, N^{(n)}$
Initialisierung Hashwerte $\vec{H}^0$
for $k \leftarrow 0$ **to** *15* **do**
  $W_k \leftarrow N_k^{(i)}$
end
for $k \leftarrow 16$ **to** *63* **do**
  $W_k \leftarrow \sigma_1(W_{k-2}) + W_{k-7} + \sigma_0(W_{k-15}) + W_{k-16}$
end
for $i \leftarrow 1$ **to** n **do**
  `/* Initialisiere mit letzten Hashwerten */`
  $\vec{a} \leftarrow \vec{H}^{(i-1)}$
  for $j \leftarrow 0$ **to** *63* **do**
    $temp_1 \leftarrow a_8 + \Sigma_1(a_5) + Ch(a_5, a_6, a_7) + \text{Konstante}_j + W_j$
    $temp_2 \leftarrow \Sigma_0(a_1) + Maj(a_1, a_2, a_3)$
    $a_8 \leftarrow a_7$
    $a_7 \leftarrow a_6$
    $a_6 \leftarrow a_5$
    $a_5 \leftarrow a_4 + temp_1$
    $a_4 \leftarrow a_3$
    $a_3 \leftarrow a_2$
    $a_2 \leftarrow a_1$
    $a_1 \leftarrow temp_1 + temp_2$
  end
  $\vec{H}^{(i)} \leftarrow a + H^{(i-1)}$
end
return $\vec{H}^{(n)}$

Algorithmus 1 : SHA-256

initiale Hashwerte (der Wurzeln der ersten acht Primzahlen in 32-Bit-Wörtern) wie auch definierte Konstanten [59] genutzt. Diese werden hier nicht explizit im Algorithmus angegeben. Für das Hashen werden unterschiedliche Funktionen genutzt, die auf Bitoperationen zurückzuführen sind. Die verwendeten Grundoperatoren für die Verarbeitung auf 32 Bitwörtern sind:

- $\otimes$ exklusives ODER,
- $\wedge$ UND,
- $\vee$ ODER,
- $\neg$ Komplement,
- $+$ Addition modulo 32,
- R^n Rechtsverschiebung von n Bits und
- Z^n zyklische Rechtsverschiebung von n Bits.

Im Algorithmus 1 werden die folgenden komplexeren Funktionen auf 32 Bitwörtern ausgeführt:

$$
\begin{aligned}
Ch(x,y,z) &= (x \wedge y) \otimes (\neg x \wedge z) \\
Maj(x,y,z) &= (x \wedge y) \otimes (xwedgez) \otimes (y \wedge z) \\
\Sigma_0(x) &= Z^2(x) \otimes Z^{13}(x) \otimes Z^{22} \\
\Sigma_1(x) &= Z^6(x) \otimes Z^{11}(x) \otimes Z^{25} \\
\sigma_0(x) &= Z^7(x) \otimes Z^{18}(x) \otimes R^3 \\
\sigma_1(x) &= Z^{17}(x) \otimes Z^{19}(x) \otimes R^{10}
\end{aligned}
$$

Eine schematische Darstellung eines Laufes innerhalb der Verschlüsselung ist in Abbildung 3.2 dargestellt. Prinzipiell wird jeder Nachrichtenblock der Länge von 512 Bit sequentiell abgearbeitet. Hierbei werden von den Initialisierungswerten die SHA-256-spezifischen Kompressionsfunktionen angewendet.

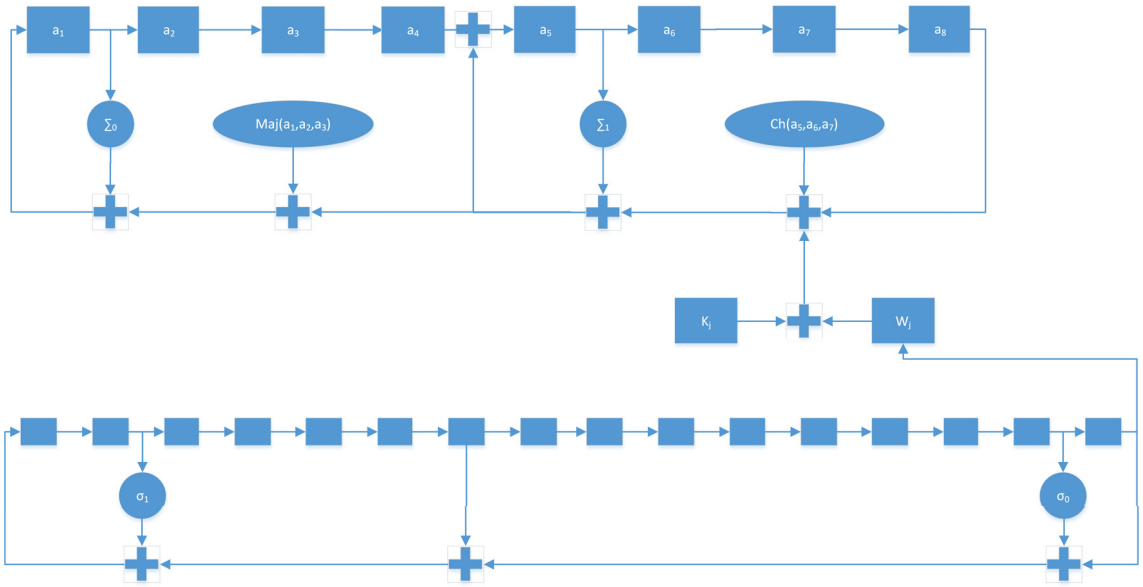

Abbildung 3.2: SHA-256-Kompressionsmethode Schritt j nach [59]

Der obere Teil der Grafik stellt die eigentliche Kompressionsmethode dar, während der untere Teil der Abbildung die Berechnung der SHA-256-Nachrichtenplanung angibt. Die SHA-256-Nachrichtenplanung korrespondiert im Algorithmus mit der Initialisierungsphase (Zeilen 10 - 12) und der anschließenden Berechnung (Zeilen 13 - 15). Diese Werte fließen in der Kompressionsfunktion dann neben den Konstanten ein (Zeile 19). Die Kompressionsfunktion für den Schritt j ist in den Zeilen 16 - 31 dargestellt. Letztlich ist das Resultat der n-te Schritt des Verfahrens und jede Ausgangsnachricht hat dieselbe Länge in der SHA-256-verschlüsselten Nachricht. Eine Rückidentifikation ist nach dem angewendeten Verfahren nicht möglich.

Aufgrund der Tatsache, dass der Hashwert und der Originalwert nicht gemeinsam zur Re-Identifikation aufgehoben werden, kann dieses Verfahren als Anonymisierungsverfahren betrachtet werden.

3.4 Assoziationsverfahren und Parametrisierung

Die Suche nach Mustern in Transaktionen, Zeitreihen und ähnlichen Konstrukten ist bereits in vielen Anwendungsfällen des Data Mining untersucht worden. Die Mustererkennung hat dabei das Ziel, Objekte zusammen zu empfehlen, die gemeinsam genutzt werden [9, 10].

Ein Ansatz, der sich in diesem Zusammenhang durchgesetzt hat, ist der Apriori-Algorithmus [7]. Hierbei werden Kandidaten erzeugt und anschließend auf den Schwellwert für die Häufigkeit geprüft. Dieser Ansatz wird in Abschnitt 3.4.1 vorgestellt. Als Alternative wurde zudem eine effizientere Datenstruktur vorgeschlagen, mit der eine Kandidatenerzeugung nicht notwendig ist. Dieser FP-Growth genannte Ansatz wird in Abschnitt 3.4.2 vorgestellt.

3.4.1 Apriori-Algorithmus

Im Folgenden wird der Apriori-Algorithmus nach Agrawal und Srikant beschrieben [7]. Hierzu wird zunächst die etablierte Notation vorgestellt, vgl. auch [97], wobei die Übertragung auf den Anwendungsfall Ausleihkontext ebenfalls erfolgt.

- Bücher (Items) $I = \{i_1, i_2, \ldots, i_m\}$ – Grundgesamtheit an Literalen
- Bücherkorb einer Person (Transaktion) $T \subseteq I$
- Datenbank D: Menge aller Transaktionen
- Support der Menge $X \subseteq T$ in D: Transaktionsanteil in D, die X enthalten:

$$support(X) = \frac{|X|}{|D|}$$

- Assoziationsregel: $A \rightarrow B$, mit $A \subseteq I$, $B \subseteq I$ und $A \cap B = \emptyset$
- Support s einer Assoziationsregel

$$A \rightarrow B \texttt{ in } D : s = support(X \cup Y) \texttt{ mit } X, Y \subseteq T$$

- Konfidenzwert c einer Assoziationsregel $A \rightarrow B$ in D: Anteil der Transaktionen, die B enthalten, wenn sie in A enthalten ist

$$c = conf(B|A) = \frac{support(A \cup B)}{support(A)}$$

Für die Eindeutigkeit des Algorithmus wird eine Sortierung der Mengen T und X durchgeführt. Die Länge einer Menge X entspricht der Anzahl k der Elemente. Somit ist die k-Elementemenge dann die Menge, die die Länge k besitzt.

Die Parametrisierung des Algorithmus nach [7] erfordert die Angaben zum minimalen Support (*minsup*) und dem minimalen Konfidenzwert (*minconf*). Der Apriori-Algorithmus lässt sich, wie in Algorithmus 2 dargestellt, formulieren.

Der Algorithmus arbeitet, indem zwei Teilprobleme gelöst werden. Hierzu werden zunächst die Körbe ermittelt, die den Anforderungen der Länge k entsprechen. Eine Initialisierung erfolgt im ersten Schritt, in dem alle möglichen ein-elementigen Körbe gebildet werden (Zeile 3). Zunächst werden alle Kombinationen von Items gefunden, die einen Support aufweisen, der mindestens so groß ist wie *minsup* (Zeilen 6-12). Nur diese Körbe werden weiter betrachtet. Nun erfolgt eine Vereinigung mit allen Elementen, die in den Körben vorhanden sind (Zeile 8). Diese werden wiederum auf ihren Support untersucht und entsprechend reduziert (Zeilen 9 - 13). Dies ist deshalb möglich, da jede Untermenge eines Korbes mit minimalem Support ebenfalls das Minimalitätskriterium erfüllt.

Anschließend werden diese Körbe in zwei disjunkte Teilmengen A und B zerlegt, so dass eine Regelerzeugung erfolgen kann. Angesichts der Tatsache, dass für das anonymisierte Empfehlungssystem nur Buchempfehlungen notwendig sind, die von einem Buch auf das nächste folgern, reicht es an dieser Stelle bereits aus, nach zwei-elementigen Mengen den Algorithmus abbrechen zu lassen. Höherwertige Regeln können keinen weiteren Beitrag an dieser Stelle leisten.

Input : Menge der Items I, Transaktionsdatenbank D, Minimaler Support minsup
Result : Menge der häufig auftretenden Muster result

```
1  C_k: zu zählende Kandidaten-Mengen der Länge k
2  L_k: Menge aller häufig vorkommenden Mengen der Länge k
3  L_1 ← I;
4  k ← 2;
5  repeat
6  |  C_k ← AprioriKandidatenGenerierung(L_{k-1});
7  |  foreach Transaktion T ∈D do
8  |  |  CT ← Subset(C_k,T);
9  |  |  foreach c ∈ CT do
10 |  |  |  c.count++;
11 |  |  end
12 |  end
13 |  L_k ← {c ∈ C_k: (c.count/‖D‖) ≥ minsup };
14 |  result ← result ∪ L_k;
15 |  k++;
16 until L_{k-1} = ∅;
17 return result;
```

Algorithmus 2 : Apriori-Algorithmus nach [7]

Für i unterschiedliche Items in den Warenkörben sind prinzipiell 2^i Itemsets möglich. Dies führt zu einer potentiellen Ergebnismenge für die Regeln von

$$\begin{aligned} \textit{Regelmenge} &= \sum_{k=1}^{i-1} \left[\binom{k}{i} \times \sum_{j=1}^{i-k} \binom{i-k}{j} \right] \\ &= 3^i - 2^{i+1} + 1. \end{aligned}$$

Aufgrund der Komplexität insbesondere bezüglich des Umgangs bei langen Sequenzen, kleinem Support oder vielen Mustern ist der Apriori-Algorithmus nicht immer geeignet. In der Literatur finden sich zahlreiche Verbesserungsvorschläge, um die Komplexität zu reduzieren. Dabei schlagen Agrawal und Srikant bereits vor, dass Transaktionen, die kein häufiges k-Itemset aufweisen, nicht benötigt werden und deshalb aus der Transaktionsdatenbank entfernt werden können [7]. Dies verbessert den Scan der Transaktionsdatenbank, ist aber zugleich mit einem Schreibaufwand verbunden. Für die Verwendung im Hauptspeicher, d.h. D befindet sich komplett im Hauptspeicher, spielt dies eine untergeordnete Rolle. Agrawal und Srikant beschreiben diesen Algorithmus als AprioriTid. Dabei ist es zudem möglich, eine Kombination beider Ansätze zu verfolgen. Dieser AprioriHybrid [7] wechselt dann vom Apriori-Algorithmus in den AprioriTid-Algorithmus, wenn die Korbmenge in den Hauptspeicher passt. Da für das Buchempfehlungssystem nur zwei-elementige Menge von Interesse sind, wäre ein Wechsel in der letzten Iteration notwendig und kann aufgrund der Wechselkosten die Vorteile nicht ausnutzen. Somit sind in Abhängigkeit der Items und Körbe sowie des Hauptspeichers nur der Alorithmus 2 und der AprioriTid sinnvoll.

Alternative Ansätze hierzu nutzen unter anderem approximative Verfahren aus. So schlagen Park et al. vor, das Zählen des Supports mittels Hash-Tabelle durchzuführen [123]. Dieses Verfahren ist schneller, aber aufgrund der nicht eindeutigen Zuordnung der Fälle in die einzelnen Buckets des Hashverfahrens ergeben sich dabei Ungenauigkeiten. Dem Verfahren liegt die Idee zugrunde, dass ein k-elementiger Korb, dessen Hash-Bucket einen geringen Zähler aufweist, nicht oft auftreten kann. Durch Verwendung einer Hash-Tabelle erfolgt somit ein besserer Zugriff.

Bei sehr großen Datenmengen empfiehlt es sich nach Toivonen nicht, mit der gesamten Datenmenge zu arbeiten, sondern nur ein Sample zu verwenden [148]. Zunächst wird ein Ausschnitt aus allen Transaktionen mit dem Apriori-Algorithmus bearbeitet. Die ermittelten Regeln werden anschließend hinsichtlich der Gesamtmenge bewertet. Je besser die Auswahl des Ausschnitts auf die Verteilung im Gesamtbestand ausfällt, desto höher ist die Güte des Verfahrens.

Srikant und Agrawal präsentieren ein Vorgehen für metrische und kategoriale Attribute [141]. Der Ansatz sieht eine Partitionierung in den entsprechenden Intervallen vor. Zudem wird das Maß der partiellen Vollständigkeit eingeführt. Damit lässt sich der Informationsverlust aufgrund der Partitionierung bestimmen. Aufgrund der vorliegenden Daten im Ausleihsystem und der Verwendung von kategorialen Daten ist dieser Ansatz nicht relevant für den Kontext dieser Arbeit, stellt aber eine interessante Erweiterungsmöglichkeit dar, um metrische Daten in die Betrachtungen zu inkludieren.

In [142] wird eine Berücksichtigung der Taxonomie auf Ebene der Items vorgestellt. Während das Anreichern der Warenkörbe mit den entsprechenden Elementen der korrespondierenden Hierarchielevel keinen hinreichend performanten Ansatz bietet, stellen Srikant und Agrawal in ihrer Generalisierung zwei Algorithmen vor, die etwa 2 bis 5 mal schneller sind als der naive Ansatz [142]. Problematisch bleibt an dieser Stelle die Generierung von zu vielen Regeln, die die Laufzeiten entsprechend negativ beeinflussen. Daher soll in der angegebenen Systemlösung zwar das Level innerhalb der Buchhierarchie eine Rolle spielen, aber die Betrachtung erfolgt gesondert, d.h. jeder Warenkorb enthält nur ein Level hinsichtlich der Taxonomie.

Im Vergleich zu Eclat [156] benutzt der Apriori-Algorithmus eine Breitensuche als Strategie. Jedoch kommt Eclat mit nur einem Datenbankdurchlauf aus. Dies ist durch eine effiziente Traversierung des geclusterten Graphen der Items möglich. Das Clustern erfolgt dabei in Äquivalenzklassen mit einer Bottom-Up Suchstrategie.

Borgelt und Kruse schlagen eine Verbesserung der Implementierung auf Basis eines Präfix-Baumes vor [32]. Dabei spielt insbesondere die Organisation innerhalb des Baumes eine entscheidende Rolle. Vor allem das Auffinden häufig genutzter Itemsets und der Speicherbedarf stehen im Fokus

des Implementierungsansatzes. Das Sortieren und die Baumaufbaustruktur haben einen essenziellen Einfluss auf die benötigte Laufzeit. Mittels der vorgeschlagenen Implementierungen können Optimierungspotentiale gehoben werden, jedoch ist dies auch abhängig von den Daten.

Zusätzlich zu den oben eingeführten Werten Support und Konfidenz werden zudem noch zwei weitere Werte in der Literatur für die Bewertung einer Regel genutzt. Diese lauten Lift und Beurteilung (engl. conviction).

Lift stellt den Zusammenhang zwischen beobachteten und erwarteten Support einer Regel dar [35]. Der Lift ist definiert als

$$lift\,(X \Rightarrow Y) = \frac{support\,(X \cup Y)}{support\,(X) \times support\,(Y)},$$

wobei die Annahme besteht, dass X und Y unabhängig voneinander sind. Die Interpretation des Lifts ist so, dass bei einem Lift von 1 Kopf und Rumpf eine unabhängige Auftrittswahrscheinlichkeit aufweisen. Das bedeutet zudem, dass bei zwei unabhängigen Ereignissen keine Regel abgeleitet werden kann. Ein Lift von größer als 1 hingegen bedeutet, dass eine Abhängigkeitsstruktur vorhanden ist und somit die entsprechenden Regeln für die Vorhersage von zukünftigen Ereignissen geeignet sind. Je höher der Lift, desto stärker ist die Regel [75]. Der Lift berücksichtigt sowohl die Konfidenz wie auch den gesamten Datenbestand.

Beurteilung einer Regel bedeutet das Auftreten des Kopfteils ohne den Rumpf in das Verhältnis gesetzt zu allen inkorrekten Vorhersagen der Regel [35]. Die Definition der Beurteilung lautet:

$$conviction\,(X \Rightarrow Y) = \frac{1 - support(Y)}{1 - confidence\,(X \Rightarrow Y)}.$$

Der Grund für die Einführung der Beurteilung einer Regel ist die Tatsache, dass die Konfidenz nicht die Richtung einer Assoziationsregel abbildet. Im Gegensatz zum Lift ist es ein gerichtetes Maß, da auch die Information zur Abwesenheit einer Rumpfregel inkludiert wird.

Neben diesen Merkmalen gibt es noch eine Vielzahl weiterer, die für die Assoziationsanalyse eine Rolle spielen. Tan et al. zeigen 21 unterschiedliche Maße, die teilweise gegensätzliche Aussagen erzeugen [146] . Jedoch wird bei einer Support-basierten Betrachtung (vergleichbar mit einer Standardisierung der Kontingenztabelle) deutlich, dass alle Maße eine hohe Korrelation aufweisen. Omiecinski führt neue Support-Werte ein: Any-Confidence, All-Confidence und Bond [122]. Während der erste Wert nicht effizient innerhalb der Mining-Verfahren anwendbar ist, können die beiden anderen effizient eingesetzt werden, da sie nach unten beschränkt sind. Ein Vergleich unterschiedlicher Maßzahlen aus dem Bereich des Machine Learnings und des Knowledge Discovery Prozesses findet sich in [95].

Zwei wesentliche Probleme ergeben sich bei der Familie der Apriori-Ansätze. Zum einen müssen eine Vielzahl von Kandidaten gebildet werden. So ergibt sich beispielsweise für 10^4 häufig auftretende ein-elementige Itemsets bereits mehr als 10^7 zwei-elementige Itemsets, die erzeugt werden müssen [79, 158]. Andererseits müssen die Kandidaten geprüft werden Dies bedeutet eine Vielzahl von Scans der Transaktionsdatenbank und der damit verbundenen Vergleiche auf Attributebene. Hier können Indexstrukturen (insbesondere Baumverfahren) zwar den Aufwand drastisch reduzieren, jedoch müssen die Vergleiche dann mit geeigneten Methoden durchgeführt werden.

3.4.2 Der Frequent-Pattern-Baum Ansatz

Die Generierung der Kandidatenmenge stellt einen hohen Aufwand dar, der in der Literatur den Apriori-Algorithmus für große Datenbestände als ungeeignet klassifiziert. Die Nachteile des Apriori-Algorithmus sind der hohe Speicherverbrauch im Hauptspeicher, so dass die Kandidatengenerierung sehr lange dauern und in Abhängigkeit der Implementierung zusätzlich Redundanzen erzeugen kann. Daher wurde von Han et al. eine neue Datenstruktur entworfen, die dieses Defizit aufhebt. Dieser Baum wird als Frequent-Pattern-Baum (FP-Baum) [79] bezeichnet.

Für die Beschreibung des FP-Baums wird im Folgenden ein Beispiel genutzt. Dazu existieren die Bücher (Items) A bis R und fünf unterschiedliche Warenkörbe. In Tabelle 3.1 sind diese abgebildet. Neben den Warenkörben wird in der dritten Spalte bereits der reduzierte und sortierte Warenkorb gezeigt, der für die eigentliche Erzeugung des Baumes verwendet wird. Hierzu muss bereits eine Vorverarbeitung durchgeführt werden.

Tabelle 3.1: Beispieldaten für FP-Growth Algorithmus

Warenkorb	Bücher	geordnete frequente Bücher
1	$\{C, A, F, G, M, I, D, H\}$	$\{C, F, A, D, H\}$
2	$\{A, B, F, C, L, D, O\}$	$\{C, F, A, B, D\}$
3	$\{B, C, P, J, O\}$	$\{C, B\}$
4	$\{B, F, K, R, H\}$	$\{F, B, H\}$
5	$\{A, C, F, E, L, H, D, N\}$	$\{C, F, A, D, H\}$

Bevor nun der Algorithmus zur Konstruktion beschrieben und das Beispiel entsprechend des FP-Baums dargestellt wird, sollen dessen Eigenschaften aufgezeigt werden. Der FP-Baum ist wie folgt definiert [158]:

- Die Wurzel wird als *leer* (null) markiert.
- Die Kinder der Wurzel werden durch Präfix-Unterbäume gebildet. Jeder Knoten eines Präfix-Unterbaumes besteht aus drei Elementen: dem Item, der Anzahl der Transaktionen in der Datenbank, die den gleichen Präfix haben und einer Sprungmarke zum nächsten Knoten mit dem gleichen Item.
- Eine Tabelle, Header-Tabelle genannt, für die Elemente der häufig auftretenden Muster, die neben dem Item die Einsprungmarke (Zeiger) zum Baum abbildet.

In Abbildung 3.3 ist der FP-Baum dargestellt, mit der sogenannten Header-Tabelle, die für jedes Item eine Einsprungsmarke (Zeiger) in den eigentlichen FP-Baum liefert. Dabei sind die Items nach ihrem Supportwert sortiert. Diese Sortierung wird beibehalten, damit gemeinsame Präfixe ausgenutzt werden können. Um eine bessere Unterscheidung zwischen dem Support, der im Intervall $[0, 1]$ definiert ist und dem Wert, der das Mindestauftreten eines Itemsets definiert zu ermöglichen, wird letzterer als Supportwert bezeichnet. Beide stehen in engem Verhältnis, da gilt:

$$support = \frac{Supportwert}{Transaktionen}.$$

Der Algorithmus zur Erzeugung des FP-Baum setzt sich aus zwei Schritten zusammen und ist in Algorithmus 3 abgebildet. In einem ersten Durchlauf der Transaktionsdatenbank wird dabei für jedes Item der Supportwert gezählt und es erfolgt eine Sortierung in Abhängigkeit dessen.

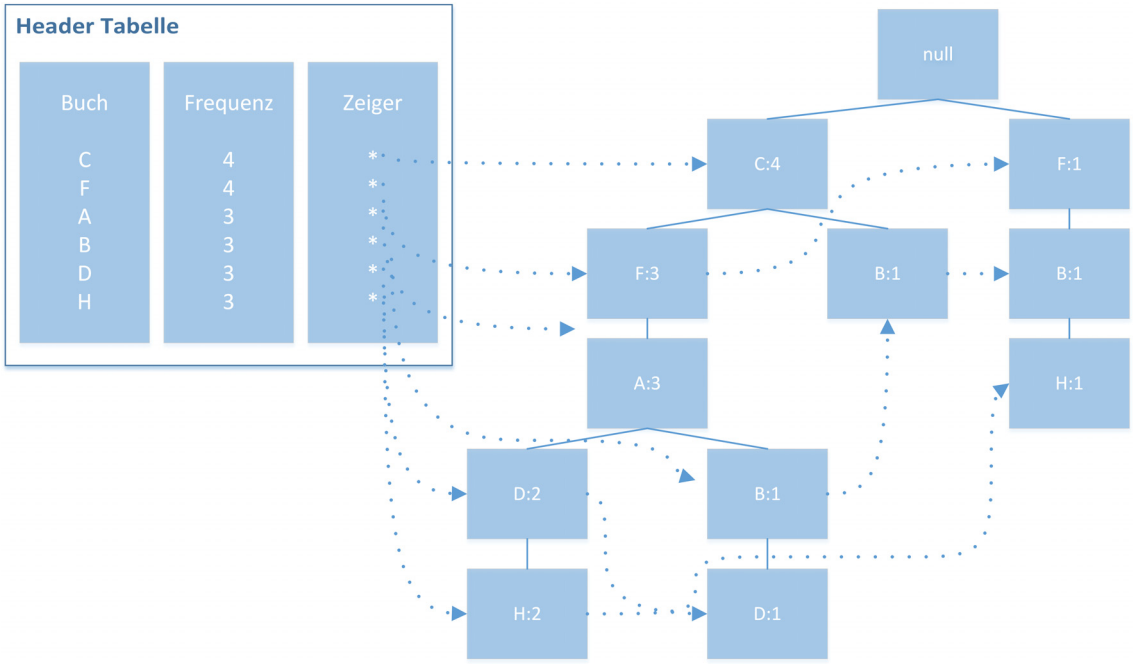

Abbildung 3.3: Frequent-Pattern-Baum nach [79]

Input : Transaktionsdatenbank D, Minimaler Support minsup
Result : FP-Baum
`/* Ermittle Support der Items */`
$L \leftarrow \emptyset$
for *alle Transaktionen* T *in* D **do**
  for *alle Items* i *in* T **do**
    L_i++
  end
end
Sortiere L absteigend
Entferne alle L_i kleiner als minsup
`/* Konstruktion des FP Baumes */`
FP-Baum $\leftarrow$ *NULL*
for *alle Transaktionen* T *in* D **do**
  Sortiere T entsprechend L
  $i \leftarrow$ ErstesElement(T)
  $I \leftarrow$ ohneErstesElement(T)
  `EinfügenInBaum` (i, I, FP-Baum)
end
return FP-Baum;

Algorithmus 3 : Frequent-Pattern-Baum-Konstruktion [79]

In Algorithmus 4 ist die Funktion des Einfügens in den FP-Baum dargestellt. Die Funktion des Einfügens in den FP-Baum arbeitet rekursiv die Elemente einer Transaktion ab, wobei die Sortierung nach der Auftrittshäufigkeit bereits erfolgt ist (Algorithmus 3 Zeile 11). Zunächst wird gesucht, ob ein gemeinsamer Pfad für die Transaktion entsprechend der Präfixdarstellung im FP-Baum existiert (Zeilen 2-8). Für diesen Fall wird der Zähler für den entsprechenden Supportwert hochgezählt (Zeile 4).

Im Fall, dass kein gemeinsamer Präfix existiert, muss ein neuer Knoten erzeugt werden (Zeile 10) und mit dem Supportwert von 1 initialisiert werden (Zeile 11). Zudem müssen die Verbindungen mittels Zeigern zu den Vorgängern (Zeile 12) und den anderen bereits vorhandenen Elementen des Items (Zeile 13) gesetzt werden.

Die Rekursion erfolgt nun durch die Prüfung, ob die noch abzuarbeitende Menge I nicht leer ist (Zeile 15). In diesem Fall wird die Einfügefunktion mit den Parametern des ersten Elementes von I, der Restmenge und dem FP-Baum aufgerufen.

Input : Erstes Element i, noch zubearbeitende Elemente I, FP-Baum
Result : FP-Baum

```
1  gefunden ← FALSE
2  for alle Kinder n in FP-Baum do
3  |  if Name(i) = Name(n) then
4  |  |  n ++
5  |  |  gefunden ⇐ TRUE
6  |  |  Beende Schleifenlauf
7  |  end
8  end
9  if gefunden = FALSE then
10 |  Erzeuge neuen Knoten n
11 |  n ← 1
12 |  Eltern(n) ← FP-Baum
13 |  Füge Zeiger entsprechend der Knotenverbinder ein
14 end
15 if I <> ∅ then
16 |  i ← ErstesElement(I)
17 |  I ← ohneErstesElement(I)
18 |  EinfügenInBaum (i, I, FP-Baum)
19 end
20 return FP-Baum;
```

Algorithmus 4 : Funktion Einfügen in FP-Baum nach [79]

Damit liegt der FP-Baum vor und kann zur Ermittlung der häufig auftretenden Muster genutzt werden. Eine Eigenschaft des Baumes ist die komprimierte Darstellung der Transaktionsdatenbank, denn voraussichtlich werden einige gemeinsame Präfixe in den Transaktionen existieren. Im besten Fall würden alle Transaktionen die gleichen Items enthalten und dies würde bedeuten, dass der FP-Baum genau einen Pfad enthält. Es kann aber auch der ungünstige Fall auftreten, dass keine Präfixe existieren, die die Transaktionen gemeinsam haben. Dann würde der Baum genauso groß sein wie die Transaktionsdatenbank, jedoch müssen noch die Speicherverbräuche für die Zeiger berücksichtigt werden. Ziel ist es nicht, den kleinsten FP-Baum zu konstruieren, sondern bei der Sortierung nach absteigender Auftrittshäufigkeit handelt es sich um eine entsprechende Heuristik.

Während der FP-Baum nun anschließend genutzt werden kann, um die Regeln abzuleiten, müssen

diese ermittelt werden, siehe hierzu Algorithmus 5. Dabei arbeitet der Algorithmus ebenfalls rekursiv, indem er Teilbäume erzeugt und somit das Gesamtproblem reduziert (Zeile 14). Für den Fall, dass nur noch ein Pfad im Baum existiert, d.h. ein gemeinsamer Präfix vorliegt (Zeilen 1-6), werden alle Kombinationen der Items gebildet und der Support aller ist das Minimum der Einzelsupports.

Andernfalls wird der Teilbaum in das Kopfelement und den Rest aufgeteilt (Zeile 8). Nun wird das Muster, dass sich aus dem Kopfelement und dem bisherigen Präfixpfad a ergibt, gewählt und bearbeitet. Der Support dieses Musters ist dabei durch den Support des Kopfelementes definiert (Zeile 10). Zur weiteren Bearbeitung muss nun der sogenannte konditionale FP-Baum für das Kopfelement a_i erstellt werden. Hierzu werden aus allen Präfixpfaden des Elementes die sogenannten konditionalen Musterpfade ermittelt. D.h. es werden alle Pfade ermittelt, die das Element a_i als unterstes Element enthalten. Auf dieser Menge aufbauend wird der entsprechende konditionale FP-Baum gebildet. Für den Fall, dass der so konstruierte Baum nicht leer ist, wird die Rekursion aufgerufen (Zeile 14).

Input : FP-Baum nach Algorithmus 3, a
Result : Menge der häufig auftretenden Muster FI

```
1  if FP-Baum hat genau einen Pfad then
2  |  for jede Kombination b der Knoten im Pfad von FP-Baum do
3  |  |  Muster m ← b ∪ a
4  |  |  Support ← min(Support_b)
5  |  end
6  end
7  else
8  |  for Jedes Element a_i im Header von FP-Baum do
9  |  |  Muster m ← a_i ∪ a
10 |  |  Support ← Support(a_i)
11 |  |  Bestimme alle Präfixpfade des Musters m
12 |  |  Erzeuge den von m ausgehenden FP-Baum_m
13 |  |  if FP-Baum_m ≠ ∅ then
14 |  |  |  FP-Growth (FP-Baum_m, m)
15 |  |  end
16 |  end
17 end
18 return FI;
```

Algorithmus 5 : FP-Growth-Algorithmus nach [79]

Die Eigenschaften des Verfahrens haben sich nach [79] sowohl als vollständig hinsichtlich der ermittelten häufig auftretenden Muster als auch nach der Korrektheit erwiesen. Ein Vergleich mit den Ergebnissen des Apriori- und des FP-Growth-Algorithmus wird in Abschnitt 5 geführt.

Eine weitere Möglichkeit der Optimierung für die Regelfindung ist der Verzicht auf eine Generierung von Kandidaten, jedoch dies direkt im relationalem Datenbanksystem, wie beim Frequent-Pattern-Growth-Algorithmus nach Shang et al. [137]. Kernidee ist „Teile und Herrsche“, so dass eine Aufteilung in kleinere Datenbanken erfolgt. Somit wird eine Kandidatengenerierung ebenso vermieden und die Transaktionsdatenbank wird in einen Baum umgewandelt. Damit erfolgt eine Reduzierung der Lesezugriffe auf die Transaktionen.

Ziel der vorliegenden Arbeit ist es, zunächst zwei-elementige Körbe zu identifizieren. Daher sind sowohl die approximativen Verfahren als auch eine Optimierung der Lesezugriffe, die insbesondere bei großen Datenbeständen eine Rolle spielen, von nachgelagertem Interesse.

3.5 Einbettung im Web-Katalog

Die Suche und Präsentation des Bestandes einer wissenschaftlichen Einrichtung erfolgt zumeist im OPAC. Aber auch neuere Dienste sind geeignet, Literaturrecherchen zu unterstützen. Für die webbasierten Dienste haben sich hier insbesondere Discovery-Systeme in den letzten Jahren erfolgreich etabliert. Zusätzlich sind auch Literaturverwaltungsprogramme oder Cloud-Angebote möglich. Da der Zugriff auf die Programmebene sich als schwierig erweist, um das Empfehlungssystem einzubetten, werden im Folgenden nur zwei Lösungen betrachtet, die den direkten Zugang zum Web-Katalog ermöglichen. Dies wird an den Beispielen der Universitätsbibliothek Magdeburg vorgestellt.

3.5.1 Der OPAC

Der Onlinekatalog stellt ein zentrales Recherche- und Präsentationsmittel des Bestandes einer wissenschaftlichen Bibliothek dar. Hierbei geht es um das Auffinden von Treffern nach bibliothekarischen Suchen. In Abbildung 3.4 ist die Startseite des OPACs der Universitätsbibliothek Magdeburg dargestellt.

Abbildung 3.4: Startseite des Benutzerkatalogs der UB Magdeburg

Die Startseite bietet zunächst einen Suchschlitz an, wobei bereits in den darüber liegenden Einstellungen deutlich wird, dass es sich um eine Boolesche Abfrage handelt. Voreingestellt sind dabei die UND-Verknüpfung der Suchwörter und dass in allen indexierten Feldern gesucht wird. Auch die Ergebnissortierung kann vorgegeben werden. Prinzipiell lassen sich auch ODER-Anfragen oder weitere Suchen wie das Stöbern im Index auswählen. Der Betreiber des OPAC kann zudem festlegen, nach welchen Metadaten gesucht werden kann. Neben Titelinformationen können dies beispielsweise Autoren oder Systematiken sein.

Da die Suche auch unter mehreren Facetten durchgeführt werden kann, stehen unter dem Reiter Erweiterte Suche komplexere Anfragen zur Verfügung. In Abbildung 3.5 ist die Erweiterte Suche aufgezeigt. An dieser Stelle wird die Mächtigkeit der Suchanfragen im OPAC deutlich. Nun kann in mehreren Suchschlitzen mit unterschiedlichen Verknüpfungen gearbeitet werden. Auch die Materialarten können für die Suchanfrage ausgewählt werden. Die erweiterte Suche wird in der Praxis häufig nur von Rechercheuren bedient, da Bibliotheksbenutzer sich mit der einfachen Suche hinreichend schnell durch den Bestand navigieren. Im Gegensatz dazu stellt die erweiterte Suche einen komplexeren Anspruch an die Handhabung des Nutzerkatalogs.

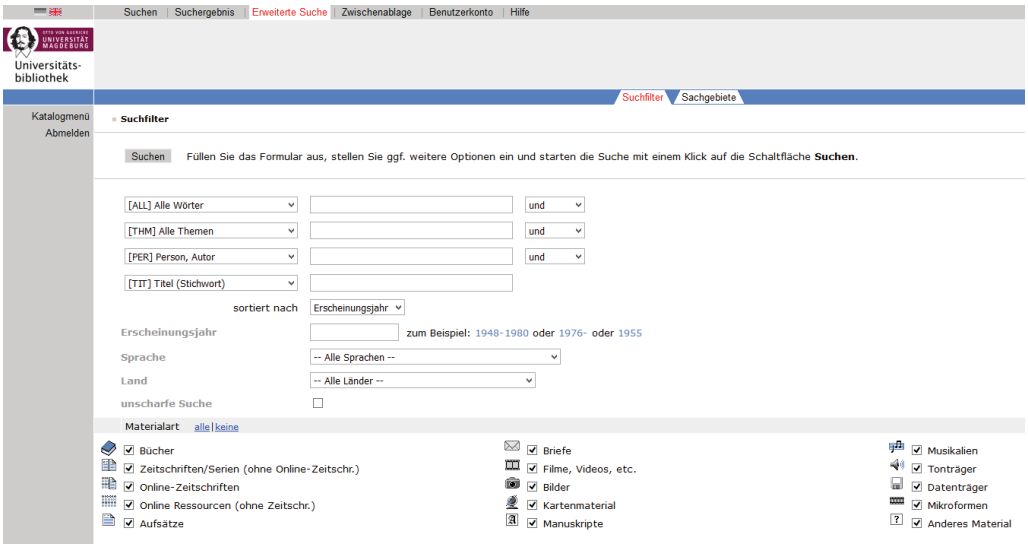

Abbildung 3.5: Erweiterte Suche im Benutzerkatalog

Ergebnis einer Suchanfrage sind häufig Trefferlisten, durch die navigiert werden muss. Dabei können durch Klick auf einzelne Treffer, die entsprechenden Daten abgerufen werden. In Abbildung 3.6 ist ein Suchtreffer dargestellt.

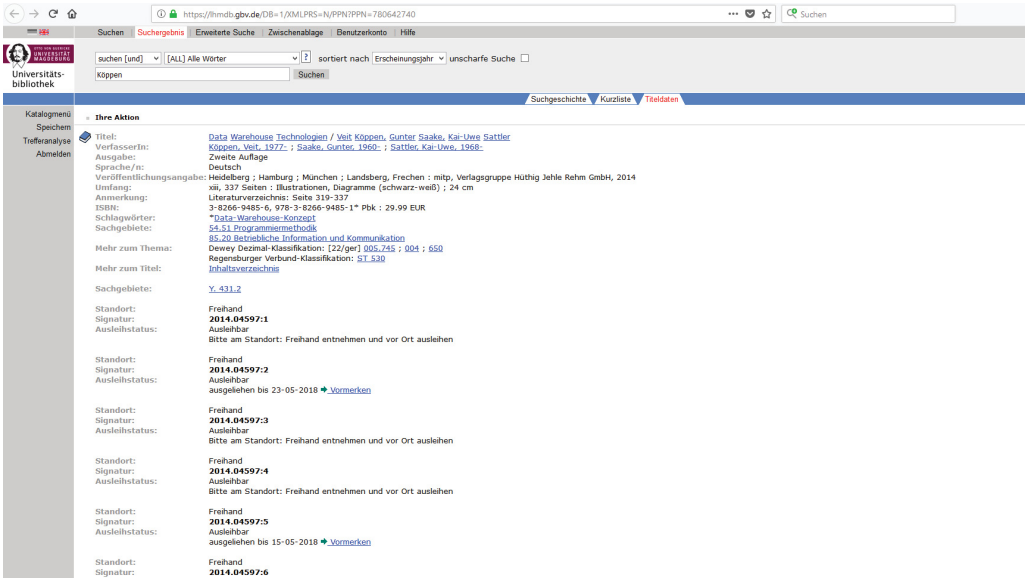

Abbildung 3.6: Ergebnisseite im Benutzerkatalog

Die Suchanfrage hierzu bezog sich auf eine Ergebnismenge von 93 Treffern. Zugleich wurde der Treffer als Zitierlink angeklickt, so dass die URL sich zusammensetzt aus der Suchanfrage an den Katalog (`https://lhmdb.gbv.de/DB=1/XMLPRS=N/PPN?PPN=`) und der PPN. Dies ermöglicht die eindeutige Identifizierung, die Weitergabe der PPN an andere eingebettete Systeme und die Weiternutzung des Links, da es sich um eine feste Adresse handelt. Gleichzeitig wird deutlich, dass die Trefferanzeige für heutige Webdarstellungen noch Platz lässt. Eine Empfehlung zum angezeigten Titel (auf PPN-Ebene) ist beispielsweise auf der rechten Seite möglich. Techniken, die dies ermöglichen, sind unter anderem Javascript-Einbettungen analog zu BibTip oder IFrames, die

andere Webseiten einbetten. Dadurch sind Anbindungen an Datenbanken oder ähnliches möglich.

3.5.2 UBfind

Der Discovery-Dienst UBfind wird seit 2015 in der UB Magdeburg produktiv eingesetzt und stellt den Prototypen des Lukida-Systems dar [61]. Die Einstiegsseite ist in Abbildung 3.7 dargestellt. Im Vergleich zum OPAC wird deutlich, dass die Suche aus Nutzersicht wesentlich einfacher gestaltet ist. Optionen sind ebenfalls möglich, aber diese sind erst durch weitere Masken zugänglich, analog zur erweiterten Suche im OPAC. Außerdem baut das Discovery-System nicht nur auf den lokalen Datenbeständen auf, sondern ist unmittelbar an den GVK angebunden. Somit besteht die Möglichkeit auch andere Bibliotheken des GBV, z.B. im Kontext der Fernleihe, einzubeziehen.

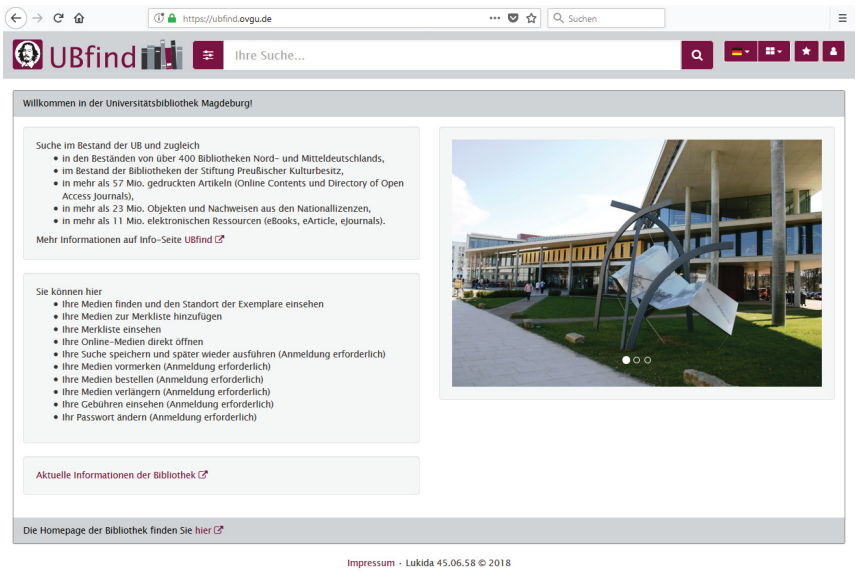

Abbildung 3.7: Startseite des Discovery-Systems UBfind

Die gleiche Suchanfrage, wie die zuvor beschriebene, wurde ausgeführt. In Abbildung 3.8 wird die gefilterte Trefferliste dargestellt. Hierbei wird deutlich, dass drei Filter eingesetzt werden können. Einerseits kann nach „Gedruckt“ oder „Elektronisch“ bzw. beiden Typen gefiltert werden, vergleiche Abbildung 3.8. Außerdem kann das Format des Werkes selektiert werden. Letztlich gibt es auch noch die Möglichkeit, den Publikationszeitraum einzuschränken. Es können noch weitere Verfeinerungen vorgenommen werden, diese sind aber aufgrund der Webseitenlogik nicht sofort zugänglich und werden daher nur selten genutzt. Der Discovery-Dienst arbeitet dabei nicht nach Boolescher Aussagenlogik, sondern es wird ein Vektorraummodell gelöst. Somit kann dies als Sortierungsproblem angesehen werden, wohingegen der OPAC ein Mengenproblem löst.

Die Trefferansicht erfolgt in UBfind in einem eigenen Overlay und ist in Abbildung 3.9 zu sehen. Neben den bibliografischen Angaben zum Titel werden in dem ebenfalls im Responsivem Design gestalteten Element die Informationen über die einzelnen Exemplare in den Kästchen hinterlegt. So wird dem Nutzer leicht ersichtlich, welche Exemplare vorhanden sind und wie deren Status ist. Dies erschwert eine direkte Verknüpfung für das Empfehlungssystem, da es nicht dazu führen soll, die einzelnen Elemente zu vermischen. Jedoch bietet die Darstellung es an, weitere Reiter zu verwenden. Für eine Empfehlung ähnlicher Suchwörter wie der gestellten Suchanfrage gibt es bereits den Reiter Ähnliche Publikationen. Dieser führt aber momentan eher zu einem Missverständnis, da

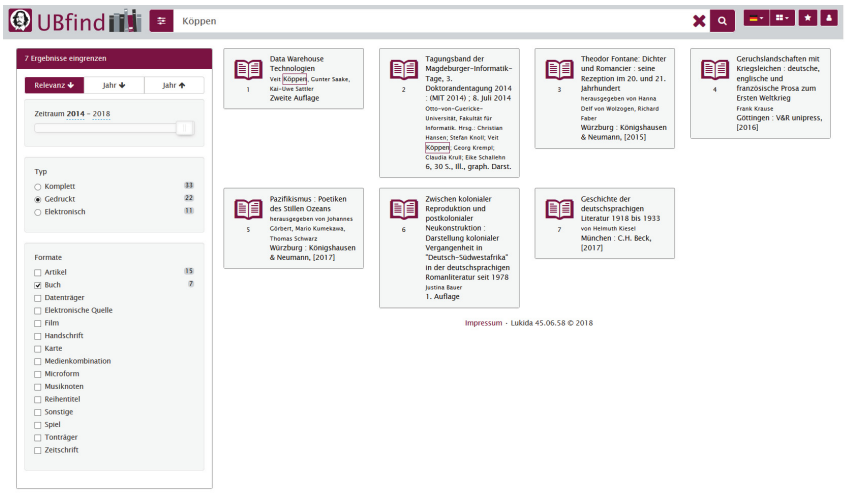

Abbildung 3.8: Ergebnisliste im Discovery-System UBfind

die Ähnlichkeit von Publikationen und nicht die der Suchanfragewörter durch Nutzer antizipiert wird.

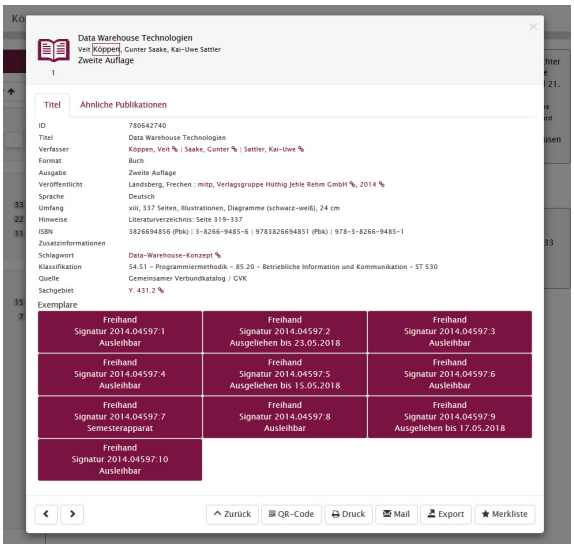

Abbildung 3.9: Ergebnisseite im Discovery-System UBfind

4 Umsetzung

In diesem Kapitel werden die Implementierungen für das Empfehlungssystem vorgestellt. Dabei wird ebenso auf die Parametrisierungen wie auch die Details der Implementierung eingegangen. Zunächst wird auf das Datenmodell des Ausleihsystems im LBS eingegangen, da dieses die Datenbasis bildet. Zugleich werden die notwendigen Transformationen beschrieben. Es folgt die Berechnung der Zusammenhänge der Verbünde. Hierbei werden sowohl der Apriori- wie der FP-Growth-Algorithmus vorgestellt. Um die ermittelten Daten in die Webrepräsentation zu überführen, müssen auch noch einige Transformationen erfolgen. Letztlich wird die Ergebnispräsentation im Webkatalog OPAC gezeigt.

4.1 Datenmodell und Transformation

Dem Ausleihsystem an der UB Magdeburg liegt das Project of Integrated Catalogue Acqusition (PICA) System LBS 4 zugrunde. Die Datenhaltung erfolgt in einem relationalen DBMS (momentan Sybase 12.5) und folgt somit dem relationalen Modell. Für die Applikationsebene ist das DBMS nicht unmittelbar erreichbar. Serverseitig existieren aber Zugriffswege, um SQL Anfragen direkt ausführen zu können. Im DBMS existieren zum aktuellen Stand 140 Tabellen und materialisierte Sichten mit insgesamt 1821 Attributen. Davon werden sowohl die Systeme für das Bestellwesen, das Ausleihsystem, die Katalogisierung und Statistiken wie die Deutsche Bibliotheksstatistik (DBS) abgedeckt. Eine Gesamtdarstellung wie auch eine komplette Darstellung für das Ausleihsystem sind daher in dieser Arbeit nicht zielführend, sondern es erfolgt eine fokussierte Darstellung des relevanten Ausschnittes.

4.1.1 Datengrundlage für die Warenkorbanalyse

In Abbildung 4.1 ist ein Ausschnitt aus dem Datenmodell des LBS 4 zu sehen, der auf den für die Warenkorbanalyse relevanten Teil reduziert ist. Bereits an dieser Stelle wird ersichtlich, dass für den Ausleihprozess eine Vielzahl an Attributen benötigt werden, die für die Warenkorbanalyse nicht von Relevanz sind. Der Warenkorb besteht im Wesentlichen aus drei Komponenten: den Nutzern, den Produkten und den daraus zusammengesetzten Transaktionen.

Die Frage, welche Attribute einen Nutzer beschreiben, die für die Warenkorbanalyse relevant sind, wird in dieser Arbeit sowohl durch den eindeutig zu identifizierenden Nutzer und einen Zeitwert beantwortet. So sind die Attribute Nutzerbarcode und Ausleihdatum hinreichend, um die Nutzer für die Warenkorbanalyse zu beschreiben. An dieser Stelle wird zusätzlich das Primärschlüsselelement (Nutzerbarcode) durch ein anderes Element aus der Menge der Kandidatenschlüssel ersetzt (der *Adress_Id* aus der Tabelle Adresse), denn damit können unmittelbare Anfragen auf Applikationsebene nicht durchgeführt werden. An dieser Stelle kann damit bereits von einer Pseudonymisierung gesprochen werden. Anzumerken ist an dieser Stelle zusätzlich, dass die Granularitätsebene des Datums, ob nun tagesgenau, monatsgenau oder auf Jahresebene, zu unterschiedlichen Attributen führt, die sich aus dem Attribut *Ausleihdatum* ableiten lassen.

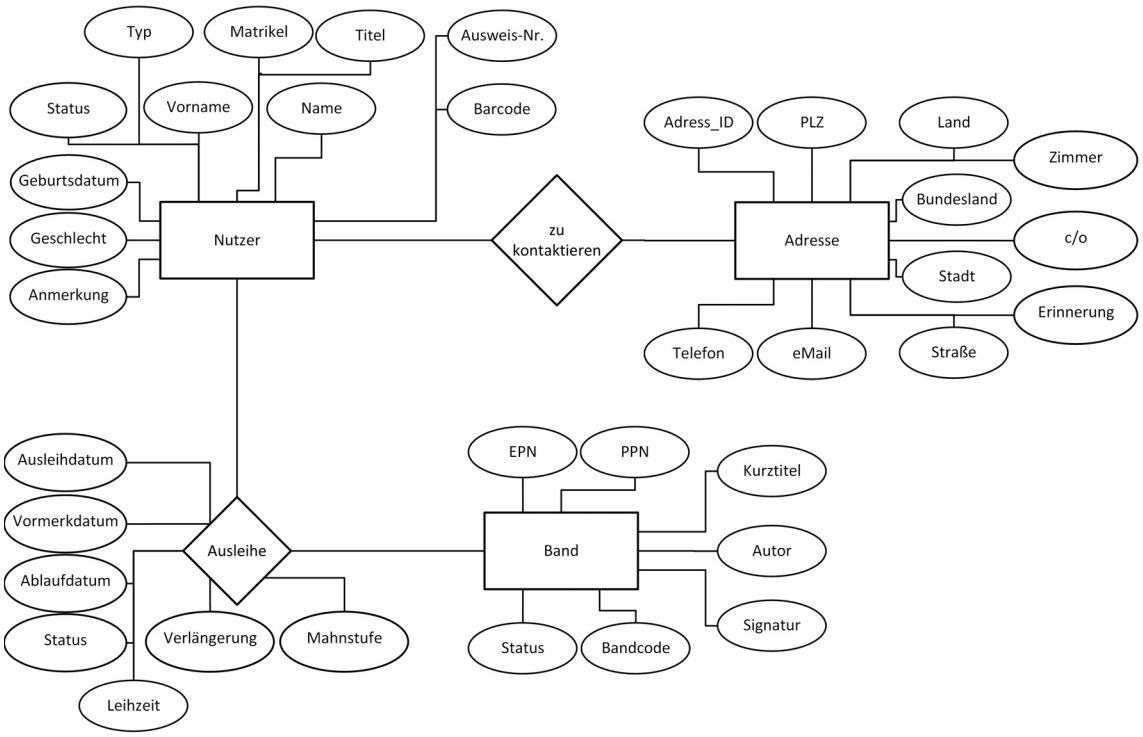

Abbildung 4.1: Auszug aus dem Entity Relationship Modell

Produkte sind im Ausleihsystem Bücher. Diese werden auf der Ebene der Exemplare durch die EPN identifiziert. Eine parallele Hierarchie beschreibt darüber hinaus Bücher. So werden alle Titel durch die PPN eindeutig identifiziert. Aber auch die Attribute Autor oder die Signatur könnten genutzt werden, um die Produkte zu beschreiben. An dieser Stelle wird für das Empfehlungssystem angenommen, dass nur ein Attribut notwendig ist. Aufgrund der Ableitbarkeit der anderen Attribute durch die EPN kann zunächst diese unmittelbar ermittelt werden. In der Datenvorverarbeitung sind dann Ersetzungen durch die entsprechende Hierarchieebene möglich. In Abbildung 4.2 wird eine mögliche Hierarchie für Bücher dargestellt.

Um die tagesaktuellen neuen Ausleihen zu ermitteln, muss somit am Ende des Tages die folgende SQL-Anfrage ausgeführt werden:

```
SELECT A.Adress_Id, B.PPN, DAY(Aus.Ausleihdatum),
   MONTH(Aus.Ausleihdatum), YEAR(Aus.Ausleihdatum)
FROM Adresse AS A, Nutzer AS N, Ausleihe AS Aus, Band AS B
WHERE A.Adress_Id = N.Adress_Id AND N.Barcode = Aus.Barcode
   AND Aus.EPN = B.EPN AND Aus.Ausleihdatum > (getdate()-1);
```

An dieser Stelle ist anzumerken, dass die Sybase-Datenbank nur die Verfahren MD5 und SHA-1 als Hashfunktionen anbietet. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) [37] empfiehlt, dass diese Verfahren nicht mehr angewendet werden sollen. Daher muss das Attribut *Adress_Id* im Sinne der kryptografischen Verfahrensempfehlung unmittelbar im Anschluss mittels einem SHA-2 Verfahren gehasht werden. Wie in Abschnitt 3.3 beschrieben, wird an dieser Stelle das SHA-256-Verfahren genutzt. Das Ergebnis der Abfrage besteht somit aus beliebigen Zeilen, die jeweils aus einem Hash und einer maximal 10-stelligen Zahl zusammengesetzt sind. Somit bildet der Hashwert die Identifikation zum Warenkorb und die 10-stelligen Zahlen bilden die Identifizierungsebene für das Buch, den Titel oder das Exemplar. Da in allen Varianten die EPN die

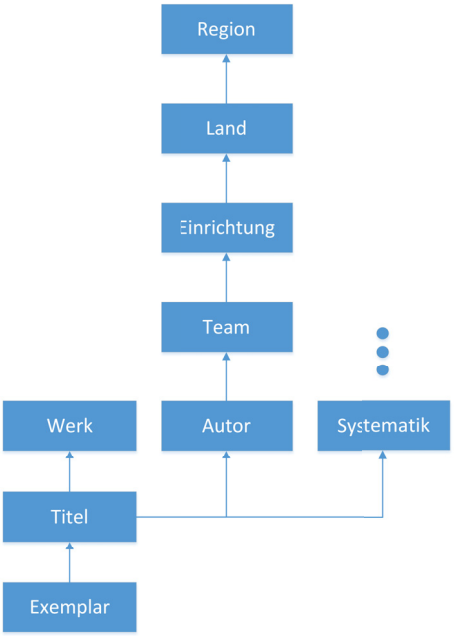

Abbildung 4.2: Parallele Hierarchie zum Produkt Bücher

granularste Ebene darstellt, jedoch unmittelbar sich auch aus der PPN die weiteren Untersuchungen leicht darstellen lassen, wird im Folgenden ohne Beschränkung der Allgemeinheit nur die PPN betrachtet. Eine Transformation in eine andere Ebene ist jederzeit möglich.

Da die eigentliche Warenkorbanalyse keinen Identifizierer für den jeweiligen Warenkorb benötigt, wird dieser dann an entsprechender Stelle nicht mehr mitgeführt. Die Verwendung eines Identifizierers ist jedoch zunächst notwendig, um aus den Einzeltransaktionen den Warenkorb zu ermitteln. In Listing 4.1 ist eine Auszug aus den Ursprungsdaten nach der Anonymisierung dargestellt.

```
1 "transaction","ppn"
2 35a9e381b1a27567549b5f8a6f783c167ebf809f1c4d6a9e367240484d8ce281_0,81341875
3 b8dc2c143be8994682b08461f46487e05874e59dd9ab65cf973e3a3c67a763aa_0,23367802
4 9a1b6288c1d0bb97708744bc0d5f778060a6aee66bd4e2abc670007bebf6f84f_0,36851916
5 edee29f882543b956620b26d0ee0e7e950399b1c4222f5de05e06425b4c995e9_0,71965228
6 48fe0661615dd0a2fc9cf1b77111613b4c3e7fc857b7bf89e472c233a0b35eb0_0,64217948
7 94f136bbfce57e5c6f1c86bbf6bf8a4e1d48ef230373eda2859170e158b1c185_0,11952986
```

Listing 4.1: Auszug aus Ursprungsdaten

4.1.2 Transformationen zur Erzeugung der Warenkörbe

Die Datentransformation erfolgt in der Programmierumgebung R [127]. In einem ersten Schritt werden die Einzeltransaktionen gelesen, siehe Listing 4.2. Das verwendete Dateiformat ist dabei eine CSV-Datei mit der gehashten Transaktion als ID für den einzelnen Warenkorb und einem Buch (Item).

Dabei werden zunächst die Transaktionen eingelesen (Zeilen 2-6), um anschließend sowohl die Transaktionsidentifizierer wie auch die Itemidentifizierer mittels Wörterbuch-Kompression, siehe hierzu [12, 26], zu transformieren. Für die Anwendung auf beide Identifizierer gibt es Gründe. Zunächst reduziert sich der Speicherverbrauch drastisch, da die Datendomäne wesentlich kleiner ist als der zur Verfügung stehende Bereich bei beiden Wertebereichen. Darüber hinaus ergeben sich sowohl

```
1  require(plyr)
2  #Dateiname festlegen
3  dateiname <- "Datei"
4  #Einzeltransaktionen der Datei laden
5  bookData <- read.csv2(file=paste0(dateiname,".csv"), sep=",", header=T)
6  names(bookData) <- c("transactionID", "item")
7  # TransactionID Mapping
8  temp <- unique(bookData$transactionID)
9  transactions <- mapvalues(bookData$transactionID, temp, seq(1,length(temp)))
10 print(paste0("Gefundene Transaktionen: ", length(temp)))
11 # Item Mapping
12 temp <- unique(bookData$item)
13 items <- mapvalues(bookData$item, temp, seq(1,length(temp)))
14 print(paste0("Gefundene Items: ", length(temp)))
15 numberitems <- length(temp)
16 write.table(cbind(temp,seq(1,length(temp))), file = paste0(dateiname,"_ItemMapping.
       csv"), sep = ",", row.names = F, quote = F, col.names = c("itemID","
       mappingItemID"))
17 bookData <- data.frame(transactions, items)
18 names(bookData) <- c("transactionID", "item")
19 write.table(bookData, file = paste0(dateiname,"_INT.csv"), sep = ",", row.names = F
       , quote = F, col.names = c("transactionID", "item"))
```

Listing 4.2: Einlesen der Transaktionen

für die Apriori-Implementierung [32] als auch für die FP-Growth-Implementierung[1] Laufzeitverbesserungen. Zudem ist anzumerken, dass insbesondere die verwendete FP-Growth-Implementierung die Wörterbuch-Kompression erfordert. Daher wird dieser Schritt der Datenvorverarbeitung notwendig.

Dies bedeutet jedoch auch, dass für die Item-Transformation die entsprechenden Werte gespeichert werden müssen, in dem sogenannten Dictionary (Zeile 16). Dies ist erforderlich, da nach Durchlauf des Algorithmus auch die entsprechenden Werte zurücktransformiert werden müssen. An dieser Stelle ist anzumerken, dass dies für die Identifizierer der Warenkörbe nicht notwendig ist. Für die Konvertierung wird auf die Methode *mapvalues* aus dem R-Paket *plyr* [154] zurückgegriffen.

Letztlich entstehen die vorverarbeiteten Daten im Integer-Zahlenformat, das in einer Datei abgespeichert wird. Ein Auszug aus einer solchen Datei ist in Listing 4.3 dargestellt.

```
1 transactionID,item
2 67,173
3 67,174
4 67,175
5 297,879
6 298,880
7 190,597
8 190,598
9 69,179
```

Listing 4.3: Auszug aus vorverarbeiteten Daten

Im nächsten Schritt werden die einzelnen Warenkörbe zusammengefasst. Dies führt zu einer weiteren Datenkompression, da für jeden Warenkorb nur noch eine Zeile verwendet wird. Während die Ursprungsdaten für den Testdatensatz B bei einer Intervalllänge von 12 Monaten ohne Überlappung 6,68 MB umfasst, weisen die vorverarbeiteten Daten noch etwa 1 MB auf. Durch die Zusammenführung der Warenkörbe reduziert sich der Umfang auf etwa 444 KB. Dies entspricht einer Kompression von etwa Faktor 15. Die Zusammenfassung der Warenkörbe wird dabei ebenfalls in R durchgeführt.

[1] https://haifengl.github.io/smile/

In Listing 4.4 ist das Vorgehen dargestellt.

```
1 trans_data <- read.transactions(file = paste0(dateiname,"_INT.csv"), rm.duplicates=
      T, format='single', sep=',', cols=c(1,2))
2 trans_dat <- DATAFRAME(trans_data, setStart = '', itemSep = ',', setEnd = '')
3 trans_dat <- trans_dat[order(trans_dat$transactionID),]
4 trans_dat <- data.frame(trans_dat$transactionID,trans_dat$items)
5 names(trans_dat) <- c("transactionID", "item")
6 trans_dat <- trans_dat[-length(trans_dat[,1]),]
7
8 write.table(trans_dat$item,file =paste0(dateiname,"_INT-baskets.csv"), sep = ",",
      row.names=FALSE, quote = FALSE, col.names = FALSE)
```

Listing 4.4: Erzeugen der Warenkörbe

Zunächst werden die vorverarbeiteten Daten eingelesen und in das Datenformat Dataframe konvertiert (Zeilen 1-2). Anschließend werden die Daten entsprechend der TransaktionsID sortiert (Zeile 3). Das Zusammenfügen der einzelnen Transaktionen erfolgt in Listing 4.4 in den Zeilen 4-6. Das Speichern der erstellten Warenkörbe erfolgt anschließend in Zeile 6. Ein Auszug der Warenkörbe mit der gleichen Parametrisierung wie oben ist in Listing 4.5 gegeben.

```
1 2741,3627,3628,3629,3630,3769,504
2 10438,27553,27554,27555,3634
3 14289,14290,2737,3635,8165
4 2645,3637,647
5 14300,24618,3646,46711,5403
6 27779,31225,33435,3652,3653,3654,3655,3656,3657,4940,49474,49475,49476
7 20533,3658,3659,7742
```

Listing 4.5: Auszug aus Warenkorbdaten

Somit liegen sowohl für den Apriori-Algorithmus wie auch dem FP-Growth-Algorithmus die Daten in einem Format vor, dass direkt genutzt werden kann.

4.2 Mustererkennung mittels Assoziationsverfahren

Die in Abschnitt 3.4 vorgestellten Verfahren zur Assoziationsanalyse können in sehr unterschiedlichen Umgebungen umgesetzt werden. Aufgrund der effizienten Umsetzung und des nahtlosen Übergangs bieten sich sowohl Implementierungen in den Umgebungen R und Java an. So kann direkt auf den vorverarbeiteten Datenbeständen gearbeitet werden und zugleich eine effiziente Verarbeitung durch die Ausnutzung des Hauptspeichers erfolgen.

Während der ursprüngliche Apriori-Algorithmus nach Agrawal und Srikant [7] hinsichtlich seiner Implementierung besonders aufgrund der Kandidatengenerierung häufig kritisiert wurde, wird in der vorliegenden Arbeit auf die Verbesserung nach Borgelt und Kruse [32] in der Implementierungsversion im R-Paket arules [77] gesetzt.

In Listing 4.6 ist der Aufruf für den Apriori-Algorithmus in der Umgebung R dargestellt. Zunächst wird das arules-Paket geladen (Zeile 1). Anschließend erfolgen die Parameterinitialisierungen für Support und die Informationen über die Warenkörbe (hinsichtlich der Transaktionen und Items), die in der Datenvorverarbeitung ermittelt wurden (Zeile 2-3). Das Einlesen der Warenkörbe wird in Zeile 4 mit der Funktion *read.transactions* bewerkstelligt. Somit liegen die Transaktionen in dem Format vor, dass der Apriori-Algorithmus direkt verarbeiten kann. Der Aufruf erfolgt in Zeile 5. Für die Parametrisierung des Algorithmus werden sowohl der Support wie auch die Konfidenz

```
require(arules)
info <- read.table(paste0(dateiname,"_trans_items.csv"), sep = ",", header = T)
supportNR <- 2
trans_data <- read.transactions(file = paste0(dateiname,"_INT-baskets.csv"), format
    ='basket', sep=',', cols=NULL)
rules = apriori(trans_data, parameter=list(support=supportNR /info$transactionNR,
    confidence=1/info$transactionNR,maxlen=2,minlen=2))
write(rules, file = paste0(dateiname,"_",supportNR,"_rules.csv"), quote=TRUE, sep =
    ",", col.names = NA)
```

Listing 4.6: Apriori-Aufruf in R

angegeben. Für die Konfidenz wird im Listing ein kleiner Wert für das einfache Auftreten bereits verwendet. Dies erzeugt alle möglichen Regeln. Die Bewertung nach der Kennzahl Lift erfolgt erst in der Nachverarbeitung der Regeldaten. Zusätzlich werden die Parameter maxlen und minlen verwendet, beide mit dem Wert 2. Dies hat zur Folge, dass nur Regeln ermittelt werden, die aus genau zwei Items bestehen, einem auf der Left-hand Side (LHS)-Regelseite und einem auf der Right-hand Side (RHS)-Regelseite. Die erzeugten Regeln folgen dem Format wie in Listing 4.7 und werden als Textdatei zur Nachbearbeitung abgespeichert.

```
"", "rules","support","confidence","lift","count"
"1","{6442} => {6443}",0.000687757909215956,1,1246.28571428571,6
"2","{6443} => {6442}",0.000687757909215956,0.857142857142857,1246.28571428571,6
"3","{23485} => {8142}",0.000687757909215956,0.857142857142857,115.041758241758,6
"4","{8142} => {23485}",0.000687757909215956,0.0923076923076923,115.041758241758,6
"5","{8641} => {7370}",0.000802384227418615,0.538461538461538,151.533498759305,7
"6","{7370} => {8641}",0.000802384227418615,0.225806451612903,151.533498759305,7
"7","{3722} => {3723}",0.000687757909215956,0.666666666666667,290.8,6
"8","{3723} => {3722}",0.000687757909215956,0.3,290.8,6
```

Listing 4.7: Auszug der Regelergebnismenge des Apriori-Algorithmus

Während die erste Spalte zur Nummerierung der Regeln dient, stellt die zweite Spalte die identifizierte Regel dar. Diese hat den Aufbau LHS => RHS, wobei es sich um Mengenkonstrukte auf beiden Seiten handelt. Da durch die Parametrisierung des Apriori-Algorithmus nur ein-elementige Regeln erzeugt wurden, ist jeweils genau eine Zahl dort in den geschweiften Klammern vertreten. Zudem werden noch der Support, die Konfidenz und der Lift als Kommazahlen geliefert. Letztlich stellt der Count die Anzahl der Warenkörbe dar, für die für die Regel gilt.

Für die Implementierung mittels FP-Growth Algorithmus wird auf die Open-Source Umgebung Smile – Statistical Machine Intelligence and Learning Engine[2] zurückgegriffen. Dabei muss beachtet werden, dass als Inputdaten nur Warenkörbe akzeptiert werden, die Items im Integerformat repräsentieren.

Somit ergeben sich zwei Implementierungsarten für die Warenkorbanalyse, die in der Evaluation (in Abschnitt 5.3.2) hinsichtlich ihrer Eignung näher beleuchtet werden.

4.3 Regelaufbereitung

Letztlich muss noch eine Transformation der Regeln erfolgen, so dass diese im Katalogsystem eingebettet werden können. Hierbei ist insbesondere die Rücktransformation in die PPN-Ebene notwendig. Zudem kann es vorkommen, dass für ein einzelnes Element sehr viele Regeln erzeugt

[2] https://haifengl.github.io/smile/association-rule.html

werden. An dieser Stelle wird dann in Abhängigkeit des Lifts nur eine Teilmenge genutzt. Hintergrund ist die damit verbundene Differential Privacy, da das Ergebnis beschnitten wird. In diesem Zusammenhang wird auch noch eine randomisierte Ausgabe in der Ergebnispräsentation (siehe hierzu Abschnitt 4.4) genutzt.

Zunächst werden die Regeln entsprechend der LHS und RHS aufgeteilt. Gleichzeitig werden die für die weitere Verarbeitung notwendigen Kenngrößen ebenfalls genutzt und in eine CSV-Datei gepackt. Die Transformation erfolgt in Java, da hier effizienter mit Zeichenketten umgegangen wird. Aufgrund der eher einfachen Transformation wird sowohl auf den Algorithmus wie auch die Ergebnisrepräsentation verzichtet, da diese zu einem ähnlichen Ergebnis führen wie in Listing 4.7.

Anschließend werden die transformierten Daten wieder in R geladen, um diese nun entsprechend der Data Privacy Anforderung aufzubereiten. Bei der Darstellung der Ergebnisse werden nur maximal drei Empfehlungen dargestellt. Daher ist es sinnvoll, auch die Anzahl der Regeln in Abhängigkeit des Lifts zu begrenzen. Im Folgenden werden für die Darstellung auf der Webseite angenommen, dass X Empfehlungen ausgesprochen werden, und in der berechneten Regelbasis maximal Y Regeln zur Verfügung stehen. Diese beiden Parameter können in Abhängigkeit der Datenbasis wie auch der anderen Parameter des Supports definiert werden. Beide Parameter dienen der Reduktion der Informationen im Hinblick auf die Differential Privacy (vergleiche hierzu insbesondere Abschnitt 2.2.2).

An dieser Stelle wird im Folgenden mit $X = 3$ und $Y = 10$ gearbeitet. Welche Parameterkombinationen sinnvoll sind, ist nicht Gegenstand dieser Arbeit, kann aber in zukünftigen Betrachtungen eine Rolle spielen.

Da von maximal zehn Werten drei Werte zufällig ermittelt werden, sollten in der Regeldatenbasis für jedes Buch auch nur maximal die zehn einflussreichsten Regeln auftreten. Daher erfolgt bereits vor dem Laden in die Datenbank eine Aussortierung der „unwichtigen“ Regeln. Dies führt zu einer deutlichen Reduktion der Empfehlungsmenge.

In Listing 4.8 ist der entsprechende R-Code dargestellt. Nachdem die Regeln geladen wurden

```
1  regel_daten <- read.csv2(file = paste0(dateiname, "_rules_forDBImport.csv"),
       sep = ",", header = T)
2
3  regel_daten <- regel_daten[with(regel_daten, order(kopf, -lift)), ]
4  anzahl_regeln <- length(regel_daten[,1])
5  anzahl_buecher <- length(unique(regel_daten[,1]))
6  buchuebersicht <- count(regel_daten[,1])
7  buchuebersicht <- buchuebersicht[with(buchuebersicht, order(-freq)), ]
8
9  for(i in 1:anzahl_buecher) {
10     indices <- which(regel_daten$kopf == buchuebersicht$x[i])
11     if(length(indices) > 10) {
12        regel_daten <- regel_daten[-indices[11:length(indices)],]
13     }
14 }
15 anzahl_pruned_regeln <- length(regel_daten[,1])
16 write.table(regel_daten, file = paste0(dateiname, "_rules_forDB.csv"), sep =
       ",", quote = F, row.names=F)
```

Listing 4.8: Trimming der Regeln

(Zeile 1), erfolgt eine Sortierung der Regeln zunächst nach dem Kopfelement und als zweite Sortierung nach dem Lift (Zeile 3). An dieser Stelle kann auch die Konfidenz oder der Support genutzt werden.

Um die Regeln hinsichtlich der Buchempfehlungen zu trimmen, erfolgt zunächst eine Übersicht

der Regeln zu den einzelnen Elementen. Hier spielt insbesondere das Auftreten der Elemente in den Regeln eine Rolle (Zeilen 4 - 7). Nun werden für jedes Buch in der Regelmenge (Zeile 9) die Einträge ermittelt. Für den Fall, dass mehr als 10 Regeln auftreten (Zeile 11) erfolgt die Eliminierung der zusätzlichen Regeln (Zeile 12). Das Ergebnis sind die getrimmten Regeln für die Webpräsentation.

Bevor das Laden der Daten in die Datenbank erfolgen kann, müssen die Items wieder mit den ursprünglichen Identifizierern aus der Datei ItemMapping.csv ersetzt werden. Dies erfolgt mit dem R-Code aus Listing 4.9. An dieser Stelle wird das plyr-Paket [154] eingesetzt (Zeile 1). Die

```
1  require(plyr)
2
3  regeln <- read.csv2(file=dateiname, sep=",", header = T)
4  mapitems <- read.csv2(file="ItemMapping.csv", sep =",", header = T)
5
6  regeln$kopf <- mapvalues(regeln$kopf, mapitems$mappingTransactionID,
       mapitems$itemID)
7  regeln$rumpf <- mapvalues(regeln$rumpf, mapitems$mappingTransactionID,
       mapitems$itemID)
8  regeln <- regeln[with(regeln, order(kopf, -lift)), ]
9
10 write.table(regeln, file = paste0(dateiname,"_",support,"_final.csv"), sep =
       ",", row.names = F, quote = F)
```

Listing 4.9: Rücktransformation der Identifizierer

Regeln und das Mapping werden in Zeile 3 und 4 eingelesen. In Zeile 6 werden alle Kopfelemente mit dem Identifizierer ersetzt und in Zeile 7 alle Rumpfelemente. Anschließend erfolgt noch eine Sortierung nach den Kopfelementen und ihrem absteigendem Liftwert. Dies ist vorwiegend der Tatsache geschuldet, dass so leichter eine Überprüfung der Elemente stichprobenartig erfolgen kann. Auch für die Indexierung in der Datenbank wird dadurch eine kleine Optimierung erreicht.

Im letzten Schritt müssen die Daten in die Datenbank geladen werden. An dieser Stelle wird MySQL genutzt. Ein mögliches Schema kann dabei Informationen zum Autor, dem Titel usw. enthalten, auch die hierarchische Beziehung kann implementiert sein, um beispielsweise mehrere Empfehlungsebenen abzudecken. Aufgrund des Fokus auf die Warenkorbanalyse soll auf eine komplette Darstellung der Einsatzszenarien verzichtet werden und an dieser Stelle nur der essenzielle Anteil dargelegt werden.

Das für die weitere Nutzung minimale Datenbank-Schema ergibt sich wie folgt:

```
CREATE TABLE RegelImport (
    Kopf INT NOT NULL, Rumpf INT NOT NULL, Support FLOAT NOT NULL,
    Confidence FLOAT NOT NULL, Lift FLOAT NOT NULL,
    INDEX KIDX (Kopf) USING BTREE, INDEX RIDX (Rumpf) USING BTREE);
```

Das Einladen der Daten kann mittels JDBC/ODBC-Treibern unter Java erfolgen oder mit anderen bzw. den mitgelieferten Tools. Bei großen Datenmengen kann es dabei erforderlich sein, effiziente Methoden wie Bulk-Loading, siehe hierzu auch [97], einzusetzen. Da diese durch unterschiedliche Tools unterstützt werden können, soll eine nähere Betrachtung nicht erfolgen. Um jedoch eine hohe Queryperformanz des Systems zu garantieren, sollte auf jeden Fall eine Indexstruktur für die Kopfelemente genutzt werden. Hier bietet sich insbesondere der B-Baum an. Bei Verknüpfungen zu anderen Relationen wie Autor oder Titelinformationen müssen zudem Indexstrukturen für die Rumpfelemente erstellt werden. Nur so kann eine gute Antwortzeit für die Webdarstellung erzielt werden. Dies wird in der Schema-Definition bereits mit den beiden Indexstrukturen erzielt. Wichtig

hierbei ist, dass beide Spalten Werte mehrfach aufnehmen können und daher weder die *UNIQUE* noch *PRIMARY KEY* Bedingungen genutzt werden dürfen.

4.4 Ergebnispräsentation im Katalogsystem

Die Ergebnispräsentation erfolgt im OPAC über die Einbindung eines iFrames. Hierzu muss mitgeteilt werden, für welchen Titel eine Empfehlung gesucht wird. In Abhängigkeit der Ergebnismenge wird dann eine Einblendung durchgeführt.

Der OPAC wird für die Titelebenen-Anzeige in diesem Zusammenhang wie in der BibTip-Lösung angepasst. Aufgrund der Realisierung im Testsystem der UB Magdeburg wird aber nicht auf eine ebenfalls mögliche Java- bzw. Javascript-Implementierung gesetzt, sondern mittels PHP: Hyptertext Preprocessor (PHP) eine Ergebnispräsentation durchgeführt.

Dazu wird zunächst der iFrame-Aufruf in der entsprechenden OPAC-Datei implementiert. Dies erfolgt durch Hinzufügen der Zeile:

<iframe src="https://<%variable(SERVER)>/empfehlung.php?PPN=<%variable(PPN)>" >.

Auf dem PHP-Webserver erledigt das in Listing 4.10 dargestellte Skript die Ausgabe.

```
1  <?php
2  require_once ('DB-Konfiguration.php');
3  $db_link = mysqli_connect (
4                          MYSQL_HOST,
5                          MYSQL_BENUTZER,
6                          MYSQL_KENNWORT,
7                          MYSQL_DATENBANK
8                         );
9  $ppn  = $_GET["PPN"];
10 echo "Ergebnisempfehlungen f&uuml;r ".$ppn;
11
12
13 $sqlAnfrage = "SELECT Rumpf FROM `RegelImport` WHERE KOPF = ".$ppn." ORDER BY RAND
       () LIMIT 3";
14
15 $dbErgebnisrg = mysqli_query( $db_link, $sqlAnfrage );
16 if ( ! $dbErgebnis )
17 {
18   die('Ung&uuml;ltige Abfrage: ' . mysqli_error());
19 }
20
21 echo '<table border="1">';
22 while ($zeile = mysqli_fetch_array( $dbErgebnis))
23 {
24   echo "<tr>";
25   echo "<td>". $zeile['Rumpf'] . "</td>";
26   echo "<td> <a href=\"https://lhmdb.gbv.de/DB=1/CMD?ACT=SRCHA&IKT=1016&TRM=ppn+".$
         zeile['Rumpf']."%3F\"> zuf&auml;llige Empfehlung </a></td>";
27   echo "</tr>";
28 }
29 echo "</table>";
30
31 mysqli_free_result( $db_erg );
32 ?>
```

Listing 4.10: PHP-Skript für die Empfehlung

Neben der Datenverbindung, die in der Datei *DB-Konfiguration.php* konfiguriert wird, erfolgt die Ergebnisrepräsentation durch das SQL-Statement aus Zeile 13. Während die Select-Anweisung die zur Präsentation wichtigen Bestandteile enthält und die Join-Beziehung sich aus dem Datenmodell

ergeben, ist die einzige Einschränkung, dass nur Ergebnisse betrachtet werden, die die PPN als Kopfelement enthalten. Zudem erzeugt die Sortierung *ORDER BY RAND()* eine zufällige Reihenfolge in der Ergebnisliste. Während im vorangegangenen Abschnitt bereits die Regelmenge pro PPN auf maximal 10 gesetzt wurde, wird mit *LIMIT 3* zugleich festgelegt, dass die Ergebnismenge maximal drei entspricht. Auf diese Weise wird eine Randomisierung im Sinne der Datenrepräsentation erreicht. Somit sind die Anforderungen im Sinne der Differential Privacy ebenfalls beachtet.

Die Uniform Resource Locator (URL) in Zeile 26 nutzt den OPAC der UB Magdeburg und setzt dabei eine Suchanfrage (CMD?ACT=SRCH) ab, die als Suchterm die PPN enthält. Um dem internen Format der PPN (Ganzzahlwert) und der PPN in der Benutzerkatalog-Sicht angereichert mit einer zusätzlichen Prüfziffer am Ende gerecht zu werden, erfolgt die Trunkierung am Ende der PPN im Suchterm. Dies ist die einfachere Lösung anstelle der Berechnung der Prüfziffer im PHP-Skript und führt zu den gleichen Ergebnissen.

Für den Fall, dass für eine PPN keine Informationen vorliegen, erfolgt kein Angebot. Prinzipiell lässt sich das Angebot auch in andere Umgebungen einbetten, z.B. den Discovery-Dienst der UB Magdeburg, bedarf an dieser Stelle jedoch einer kleinen Änderung, die dann durch die Systemumgebung realisiert werden muss. Aufgrund der Implementierungsdetails wird auf eine Wiedergabe an dieser Stelle verzichtet.

5 Evaluation der Lösung

Die im vorhergehenden Kapitel beschriebene Umsetzung soll an dieser Stelle evaluiert werden. Aufgrund der Parametervielfalt werden zwei Testdatensätze genutzt, die aus einem künstlich erzeugten Datensatz mit 330 Buchungen und einem Datensatz aus dem LBS System der UB Magdeburg bestehen. Das Evaluierungskapitel ist so aufgebaut, dass die Forschungsfragen am Ende beantwortet werden. Somit werden unterschiedliche Aspekte, wie Parameterwerte und Einflüsse dieser untersucht. Die quantitative Untersuchungen werden zumeist ceteris paribus durchgeführt. Jedoch sind aufgrund der Komplexität einige Punkte dahingehend schwieriger in der Analyse. Somit wird mit unterschiedlichen Parametersätzen die Präsentation durchgeführt.

Zunächst erfolgt eine Beschreibung der Testumgebung. Danach werden die verwendeten Datensätze vorgestellt. In der Analyse werden insbesondere die drei Schritte: Warenkorbbildung, Einfluss des Supports und Trimming der Regeln betrachtet.

Für die Forschungsfragen sind zusätzlich die Resultate der Verwendung unterschiedlicher Hierarchiestufen und das Entfernen älterer Information von Belang. Zudem wird der reale Datensatz noch in Bezug auf seine Güte hinsichtlich späterer Ausleihen untersucht. Hierzu dienen die Ausleihen des Monats nach Ende der Warenkorberstellung. Dieses Kapitel schließt mit einer Zusammenfassung hinsichtlich der Forschungsfragen.

5.1 Beschreibung der Testumgebung

Mit der Umstellung von LBS-3 auf LBS-4 stellt die Verbundzentrale des GBV neben dem Produktivsystem auch ein Testsystem zur Verfügung, dass die gleichen Funktionalitäten erfüllt und insbesondere bei Änderungen genutzt werden kann. Es bietet sich daher an, dieses System als Basis für die Implementierung und Evaluation zu verwenden. Der Testdatensatz A wurde daher auch im Testsystem komplett erarbeitet. Aufgrund der notwendigen realen Datensätze wurde der Testdatensatz B hingegen aus dem Produktivsystem ermittelt. Dabei wurde bereits die reibungslose Überführung festgestellt, so dass eine Implementierung der Empfehlungslösung ohne großen Zusatzaufwand möglich ist.

Für die Durchführung von Datenvorverarbeitung, Data Mining und Datennachbereitung wurden sowohl Java-Komponenten wie auch R-Skripte verwendet. Da das zugrundeliegende Betriebssystem ein UNIX-Derivat darstellt, sind beide Programmierumgebungen nutzbar.

Die Ergebnisrepräsentation wird im OPAC mittels iFrame umgesetzt, so dass an dieser Stelle ein eigener Webserver mit PHP und MySQL benötigt wird. Somit kann einerseits mittels virtuellem Serverkonzept eine Entkoppelung erzielt werden, ohne neue Hardware zu benötigen. Andererseits ist falls notwendig eine Systementlastung ebenfalls leicht realisierbar.

Auswahl und Vergleich der Algorithmen

In Abschnitt 3.4 wurden zwei Verfahren vorgestellt, die in Abschnitt 4.2 als Implementierung vorgeschlagen wurden. In Abhängigkeit der Parametrisierung werden sowohl die Laufzeit der

Algorithmen wie auch der Speicherbedarf gesetzt. An dieser Stelle ist anzumerken, dass die Apriori-Implementierung ein stabileres Verhalten aufweist, denn bei geringen Supportwerten reicht der durch die Virtuelle Maschine (VM) bereitgestellte Heap-Speicher nicht aus. Daher wird deutlich, dass obwohl keine Kandidatengenerierung beim FP-Growth erfolgt, die Rekursion bzw. das Traversieren der Regeln einen sehr hohen Speicherbedarf hat. Problematisch ist an dieser Stelle insbesondere, dass, obwohl mehr Hauptspeicher zur Verfügung steht, der Heap-Speicher die beschränkende Ressource darstellt.

Für den Fall, dass ausreichend Speicher zur Verfügung steht, liefern beide Implementierungen die gleichen Lösungen. Da der Supportwert der entscheidende Faktor bei der Laufzeitbestimmung ist, wird in Abschnitt 5.3.2 eine kurze Darstellung der Laufzeiten im Vergleich beider Algorithmen durchgeführt. Zunächst erfolgt eine kurze Beschreibung der Testfälle.

5.2 Beschreibung der Testfälle

Für die Evaluation werden zwei unterschiedliche Testdatensätze verwendet. Während für die Forschungsfragen die prinzipielle Evaluation anhand eines künstlich erzeugten Testdatensatz A erfolgt, soll mit Testdatensatz B die Einsatzfähigkeit unter realen Bedingungen geprüft werden.

5.2.1 Testdatensatz A

Der erste Testdatensatz wurde im LBS System mit 330 Buchungen erzeugt. Hier wurden neben Ausleihen und Rückgaben auch Vormerkungen und Verlängerungen sowie Magazinbestellungen genutzt. Insgesamt wurden 47 unterschiedliche Bücher auf Ebene der Exemplare verwendet. Für die Buchungen wurden drei Nutzergruppen aus den Fachgebieten Ingenieurswissenschaft (10 Nutzer), Psychologie (9 Nutzer) und Informatik (10 Nutzer) genutzt. Zudem wurden einzelne Bücher disziplinübergreifend ausgeliehen und es gibt drei Nutzer, die keiner Gruppe zugeordnet werden. Dies ergibt einen Gesamtbestand von 47 Büchern bei 32 unterschiedlichen Nutzern. Die Verbuchungen wurden so gewählt, dass einzelne Buchungszeiträume einen Ausleihzeitraum darstellen. In Summe wurden 45 Zeitabschnitte definiert, die für die weiteren Untersuchungen genutzt werden sollen.

In der Tabelle 5.1 sind die verwendeten Exemplare dargestellt. Für die Auswirkungen der Buchempfehlungen und die Darstellung innerhalb der Hierarchie sind sowohl die EPN, wie auch PPN und das Themengebiet dargestellt. Die Titelebene wird durch den Buchtitel repräsentiert, auch wenn an dieser Stelle explizit die Auflagen aufgrund der besseren Darstellungsweise nicht angegeben sind.

Mit der Zuordnung zu den Autoren ergeben sich ebenfalls Möglichkeiten, diese als Empfehlungsgrundlage zu nutzen. An dieser Stelle muss jedoch dann, aufgrund der fehlenden Verknüpfungen im OPAC, ein Umweg über die Gemeinsame Normdatei (GND) der Deutschen Nationalbibliothek (DNB) gefunden werden. In dieser Arbeit wird dieser Weg nicht weiter verfolgt, obwohl er prinzipiell umsetzbar erscheint.

Eine weitere Anmerkung betrifft die EPN und PPN. Beide werden ohne die letzte Ziffer (Prüfziffer) angegeben. Diese lässt sich aber ermitteln bzw. die verkürzte Darstellung führt bei Anwendung der Trunkierung innerhalb der Suchanfrage stets zu einem eindeutigen Treffer im OPAC.

5.2.2 Testdatensatz B

Der zweite Datensatz für die Evaluation wurde unmittelbar aus dem LBS der Universitätsbibliothek Magdeburg erzeugt. Hierbei handelt es sich um alle Ausleihen und Bestellungen, die im Zeitraum 1.

Tabelle 5.1: Bücherliste für Testdatensatz A

EPN	PPN	Fachgebiet	Titel	Autoren
287979086	02758258	Psychologie	Lehrbuch allgemeine Psychologie	Spada
74635137	37690306	Psychologie	Lehrbuch allgemeine Psychologie	Spada
174432326	87652952	Psychologie	Lehrbuch allgemeine Psychologie	Spada
26914332	02562166	Psychologie	Entwicklungstheorien	Flammer
23065563	21463258	Psychologie	Entwicklungstheorien	Flammer
33854792	21463258	Psychologie	Entwicklungstheorien	Flammer
21871956	21463258	Psychologie	Entwicklungstheorien	Flammer
127139367	56894412	Psychologie	Entwicklungstheorien	Flammer
94064253	56894412	Psychologie	Entwicklungstheorien	Flammer
172390684	88941834	Psychologie	Entwicklungstheorien	Flammer
134723124	17879250	Psychologie	Der Sinn der Reifungsstufen	Schmeing
133297030	19055823	Psychologie	Entwicklungspsychologie des Grundschulkindes	Kroh
51907841	33520286	Psychologie	Anfänge der Reifezeit	Stern
63663015	13306435	Informatik	Coding theorems of information theory	Wolfowitz
10405715	02408542	Informatik	Temporal logic of programs	Kröger
13553046	02408542	Informatik	Temporal logic of programs	Kröger
13553047	02408542	Informatik	Temporal logic of programs	Kröger
19077359	19553587	Informatik	Mathematics and computers	Stibitz; Larrivee
165754922	47241116	Informatik	Mathematik für Informatiker	Hachenberger
165755113	55944543	Informatik	Mathematik für Informatiker	Hachenberger
119234471	55944543	Informatik	Mathematik für Informatiker	Hachenberger
91350909	55058715	Informatik	Theoretische Grundlagen der Informatik	Socher
92668487	58717396	Informatik	Mathematik für Informatiker	Kreußler; Pfister
83486690	54371816	Informatik	Analysis und Statistik	Teschl; Teschl
63524617	31836999	Ingenieurwesen	Thermodynamik	Windisch
63524616	31836999	Ingenieurwesen	Thermodynamik	Windisch
63524615	31836999	Ingenieurwesen	Thermodynamik	Windisch
148108237	77250917	Ingenieurwesen	Thermodynamik	Windisch
128698023	66173554	Ingenieurwesen	Thermodynamik	Windisch
128698022	66173554	Ingenieurwesen	Thermodynamik	Windisch
128698020	66173554	Ingenieurwesen	Thermodynamik	Windisch
15698475	15723418	Ingenieurwesen	Grundlagen der technischen Thermodynamik	Doering; Schedwill
88057886	54332004	Ingenieurwesen	Grundlagen der technischen Thermodynamik	Doering; Schedwill; Dehli
72477680	37342198	Ingenieurwesen	Grundlagen der technischen Thermodynamik	Doering; Schedwill; Dehli
23369047	14290446	Ingenieurwesen	Thermodynamik	Schottky; Ulich; Wagner
17226869	14450400	Ingenieurwesen	Technische Wärmelehre	Lorenz
169990066	88141092	Ingenieurwesen	Thermodynamik für Dummies	Gerl; Ruderich
162375776	82618114	Ingenieurwesen	Thermodynamik	Lauth; Kowalczyk
83500505	34487131	ohne	Die Schöpferin von Harry Potter	Smith
172482506	87887891	ohne	Zellbiologie	Plattner; Hentschel
172341185	87887891	ohne	Zellbiologie	Plattner; Hentschel
172394169	89546802	ohne	Informationskompetenz	Sühl-Strohmenger; Barbian
123403983	48231965	ohne	Reiten	Barth
172446436	87288715	ohne	Thermodynamics	Duroudier
172391965	84617692	ohne	Emotionale Kompetenz bei Kindern	Petermann; Wiedebusch
142775718	72853526	ohne	Ordnungen, Verbände und Relationen mit Anwendungen	Berghammer
170777603	89569578	ohne	Main memory database systems	Faerber et al.
165833877	88169514	ohne	Angst	Micali; Fuchs
124830053	17773989	ohne	Der Ingenieurberuf	Lorenz

März 2017 bis 28. Februar 2018 angefallen sind. In diesem Zeitraum wurden durch 8724 Nutzer 51823 unterschiedliche Titel ausgeliehen mit insgesamt 91217 Transaktionen im Ausleihkontext. Die Dateien werden verschlüsselt nach neuesten Standards abgelegt, wie in [147] beschrieben.

Es ist anzumerken, dass im Gegensatz zum Testdatensatz A hier direkt nur die PPNs aufgenommen wurden, so dass ein Rückschluss auf die Ebene der Exemplare nicht möglich ist. Da Ausleihen in der UB Magdeburg nur erfolgen können, wenn diese geöffnet ist, ergeben sich für die Schließtage keine entsprechenden Buchungsvorgänge. Daher ist der Datenbestand auf 317 Dateien im Einzeltransaktionskontext für diesen Datensatz gegeben. Die Ausleihdaten wurden auf Tagesbasis erhoben.

5.3 Analyse der Testfälle

Die Analyse der Testfälle ergibt sich einerseits aus den im Kapitel 4 beschriebenen Vorgehen und andererseits mit dem Fokus auf die Forschungsfragen. Es soll an dieser Stelle das Experimentaldesign vorgestellt werden.

Für die Bildung der Warenkörbe existieren zwei Parameter, die von Interesse sind:
(1) Zunächst ist das Zeitintervall zu definieren, in dem von einem Warenkorb gesprochen werden kann. (2) Darüber hinaus kann es sinnvoll sein, eine Überlappung der Items in den Warenkörben zu ermöglichen. Dies wird mittels Parameter Überlappung festgelegt. Für die weitere Evaluation werden unterschiedliche Warenkörbe anhand dieser beiden Parameter erstellt. Die Erstellung erfolgt mittels Java Programm, das die wie in Listing 5.1 dargestellte Transformation übernimmt.

```
1  anzahlMonate = ((maxJahr - minJahr) * 12 + (maxMonat - minMonat)) + 1;
2  temp = initialisiereMonat();
3  for (int j = 0; j < (anzahlMonate / (this.intervall - this.overlap)); j++) {
4    temp_nextMonatStart = this.getNextPeriodStart(temp);
5    System.out.println("Warenkorb " + j + " wird bearbeitet. Startmonat: " +
         temp);
6    for (int i = 0; i < this.interval; i++) {
7      System.out.print(i + " ");
8      this.datenLesen.alleDateienEinlesen(this.dateiNamenStart, temp);
9      temp = getNextMonat(temp);
10   }
11   this.datenLesen.writeTransactionsToFileAppendMode(this.resultDatei, j);
12   System.out.println("");
13   temp = temp_nextMonatStart;
14 }
```

Listing 5.1: Erstellen der Warenkörbe

Für die Warenkorberstellung spielen die Parameter *intervall* und *overlap* die zentrale Rolle. Während der erste Parameter die Anzahl der gemeinsamen Monate angibt, legt der zweite Parameter fest, wie groß der Bereich der Überlappung ist. Somit muss *overlap* stets kleiner sein als *intervall*. In Zeile 1 wird die Gesamtanzahl aller Monate ermittelt, die sich aus der Datenbasis ergeben. In Zeile 2 erfolgt die Initialisierung des Startmonats.

Zwei Schleifen sind verantwortlich für das Abarbeiten der Datenbasis. Während die äußere Schleife (mit Laufvariable j) für die Zuordnung zu den Warenkörben verantwortlich ist, kümmert sich die innere Schleife (mit Laufvariable i) um die Zusammenführung der Datenbasis auf Monatsebene (Zeile 8) in die Warenkörbe. Wenn ein Warenkorb abschließend gefüllt ist, wird dieser in die entsprechende

Resultatdatei geschrieben (Zeile 11). Grundannahme hierbei ist, dass die Datenbasis auf Dateiebene vorliegt und mittels der Methode *alleDateienEinlesen* die Zuordnung eines Monats erfolgen kann.

Für die Evaluation werden jedoch zahlreiche unterschiedliche Parametrisierungen benötigt, so dass eine Aussage zur geeigneten Parameterwahl bzw. deren Einflüsse auf die Resultate erfolgen kann. Aufgrund der bereitstehenden Datenbasis von 45 Monaten im Testdatensatz A wurden die folgenden unterschiedlichen Parameter genutzt, um entsprechende Warenkörbe zu erzeugen:

$$intervall = \{12, 18, 24, 30, 36\}$$
$$overlap = \{0, \ldots, 12\}\,.$$

Damit ergeben sich 64 unterschiedliche Warenkörbe, da für den Parametersatz (12,12) kein Warenkorb erzeugt wird.

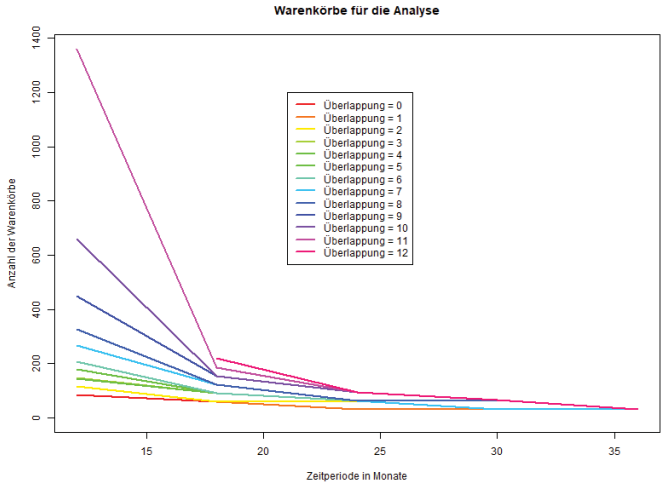

Abbildung 5.1: Ermittelte Warenkörbe für Testdatensatz A

Wie viele Warenkörbe erzeugt werden ist von den beiden Parametern abhängig. Die Darstellung erfolgt in Abbildung 5.1.

Mit steigender Überlappung steigt auch die Anzahl der Warenkörbe an und mit geringerer Periodizität ergibt sich ebenfalls eine Erhöhung der Warenkörbe. Somit kann ein quadratischer Zusammenhang zwischen beiden Parametern in Bezug auf die Warenkorbanzahl bestätigt werden.

Da kein Datensatz (12,12) erzeugt werden kann, weist der Datensatz (12,11) die größte Anzahl an Warenkörben auf. Hierbei wird ein 12-Monatelanges „Fenster“ von Monat zu Monat geschoben und erzeugt jeweils einen neuen Warenkorb. Es ergeben sich aus der Datenbasis somit 1361 Warenkörbe. Für den Fall, dass man die Realität exakt abbilden möchte, muss die volle Länge des Zeitintervalls genutzt werden und zugleich eine Überlappung ausgeschlossen werden. An dieser Stelle ist die Anzahl der Warenkörbe dann die gleiche wie die Nutzeranzahl (32).

Jedoch bedeutet dies im Sinne der Privacy ein hohes Risiko der Rückidentifikation und zugleich für die Warenkorbanalyse eine eher geringe Datenbasis.

Auf Basis der PPN als Itemgrundlage existieren in den Ausgangsdaten 38 Items. Mit der Erstellung der Warenkörbe sind jedoch nicht immer alle Items in den Warenkörben vertreten. In Abbildung 5.2 werden die unterschiedlichen Parametrisierungen und ihr Einfluss auf die berücksichtigten Bücher dargestellt.

An dieser Stelle wird deutlich, dass mit einer geringeren Länge des zu berücksichtigenden Zeitraums

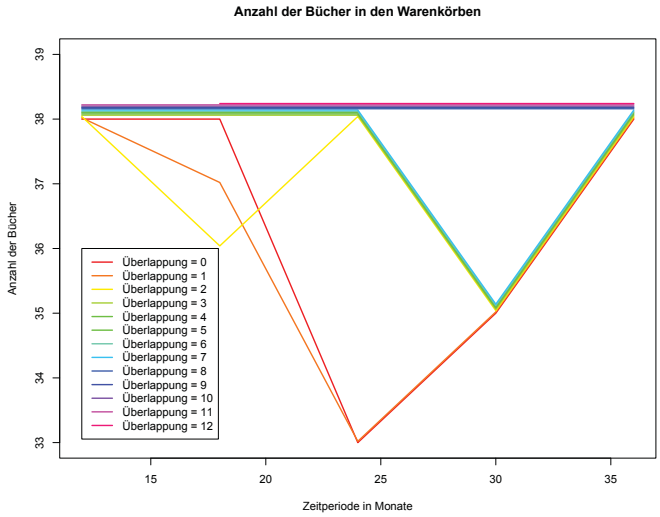

Abbildung 5.2: Ermittelte Bücher in den Warenkörben für Testdatensatz A

auch das Risiko einhergeht, dass einige, die neueren Buchungen betreffende Bücher, nicht inkludiert sind. Ergibt sich ein großer Restzeitraum, der nicht in der Warenkorberstellung berücksichtigt wird wie z.B. bei der Zeitraumlänge von 24 Monaten bei keiner oder nur einer Überlappung ist die Buchanzahl am geringsten. Hier werden nur 33 der 38 Items berücksichtigt. Mit steigender Überlappung entfällt dieses Problem, so ist es bei einer Überlappung von neun Monaten nicht mehr vorhanden.

Besonders hervorzuheben ist an dieser Stelle die Tatsache, dass bei geeigneter Wahl des Intervalls (z.B. 12 oder 36) die Überlappung keinen weiteren Einfluss auf die Anzahl der Bücher hat.

Für den Testdatensatz B wurden folgende Parameter evaluiert und führen jeweils zu den entsprechenden Dateien der Einzeltransaktionen bzw. Warenkörbe:

Für den gesamten Zeitraum von zwölf Monaten ergibt sich ein einheitlicher Datensatz für die ersten Werte (von zwölf bis acht Monaten). Danach ist in Abhängigkeit der Überlappung (mit steigender Überlappung nehmen die Warenkörbe zu) ein Anstieg in den Warenkörben zu verzeichnen. Anzumerken ist dabei, dass aufgrund der Tatsache, dass bei einer Periode von sechs Monaten alle Transaktionen der zwölf Monate verwendet werden, jedoch bei einer Periode von fünf und Überlappung von 0 nur zehn der zwölf Monate verwendet werden. Daher ist die Anzahl der Warenkörbe, wie in Abbildung 5.3 ersichtlich, bei fünf Monaten etwas geringer. Nichtsdestotrotz steigt die Anzahl der verfügbaren Warenkörbe mit kleinerer Länge des Intervalls, da insbesondere die Werte kleiner als fünf keinen Restzeitraum offen lassen, der in der Warenkorberzeugung nicht berücksichtigt wird.

Somit kann auch für den Datensatz über die realen Buchungsvorgänge ein ähnliches Verhalten, wie im Testdatensatz A beschrieben, beobachtet werden.

5.3.1 Die Warenkörbe

Zunächst sollen die Warenkörbe näher beleuchtet werden. Dabei bilden zwei Parameter die Ausgestaltung der Warenkörbe.

In den Abbildungen 5.4 bis 5.9 sind die Histogramme der Warenkörbe hinsichtlich der beinhalteten Itemanzahl abgebildet. Diese unterscheiden sich nur geringfügig voneinander. Für eine Intervalllänge

Tabelle 5.2: Warenkorbparameter für den Testdatensatz B

Periode	Überlappung	Periode	Überlappung
12	0	12	1
11	0	11	1
10	0	10	1
9	0	9	1
8	0	8	1
7	0	7	1
6	0	6	1
5	0	5	1
4	0	4	1
3	0	3	1
2	0	2	1
1	0	12	2
12	3	11	2
11	3	10	2
10	3	9	2
9	3	8	2
8	3	7	2
7	3	6	2
6	3	5	2
5	3	4	2
4	3	3	2

von zwölf entstehen etwa zweieinhalb Mal so viele Warenkörbe (83) im Vergleich zum Gesamtintervall (32). Zu beachten ist jedoch, dass mit steigender Intervalllänge l die Anzahl an Warenkörben sinkt. Durch eine Verkürzung des Betrachtungszeitraumes kommen zwar weniger Elemente in die Körbe, jedoch entstehen mehr Warenkörbe, da diese durch die Aufteilung vervielfacht werden. Während im Gesamtzeitraum nur zwei Warenkörbe mit sechs Elementen vorhanden sind, können in allen anderen auch Warenkörbe mit fünf Items beobachtet werden. Die Anzahl der kleinen Warenkörbe steigt mit sinkender Intervalllänge. Die Warenkörbe mit nur einem Element können für die Regelerzeugung nicht weiter genutzt werden. Dies bedeutet, dass mit zu kurzer Intervalllänge Beziehungen aufgebrochen werden, die jedoch für die Warenkorbanalyse wichtig sind. Mit dem zweiten Parameter, der Überlappung, soll noch geprüft werden, inwieweit die Beziehungen erhalten werden können.

Zunächst erfolgt aber noch die Betrachtung der relativen Itemhäufigkeit in den Warenkörben. Die erfolgt wiederum für die Intervalllängen $l \in [12, 18, 24, 30, 36, 44]$ in den Abbildungen 5.10 bis 5.15.

Leichte Veränderungen sind in den Abbildungen für die Itemhäufigkeit sichtbar, wobei in den Darstellungen die relative Itemhäufigkeit genutzt wird. Zur besseren Übersichtlichkeit wurden in allen Abbildungen nur Elemente ausgewählt, die einen Support von mindestens fünf Prozent aufweisen. Das häufigste Element ist die 21, mit sechs Repräsentationen im Warenkorb bei Betrachtung des Gesamtzeitraumes. Dies entspricht einer relativen Itemhäufigkeit von 18,75 Prozent. Mit kleinerer Intervalllänge l ergeben sich kleinere relative Häufigkeiten, zugleich erfüllen weniger Elemente den Schwellenwert. Somit wird deutlich, dass mit verkürzter Intervalllänge die Zusammenhänge sich zwar reduzieren, aber nicht auflösen.

Letztlich soll exemplarisch an den Dendrogrammen für die beiden Intervalllängen $l = 12$ und $l = 44$ der Einfluss der Intervalllänge abgebildet werden. Diese sind in Abbildung 5.16 und 5.17 visualisiert.

Für die Darstellung wurde als Ähnlichkeitsmaß die Jaccard-Distanz verwendet. Dabei werden

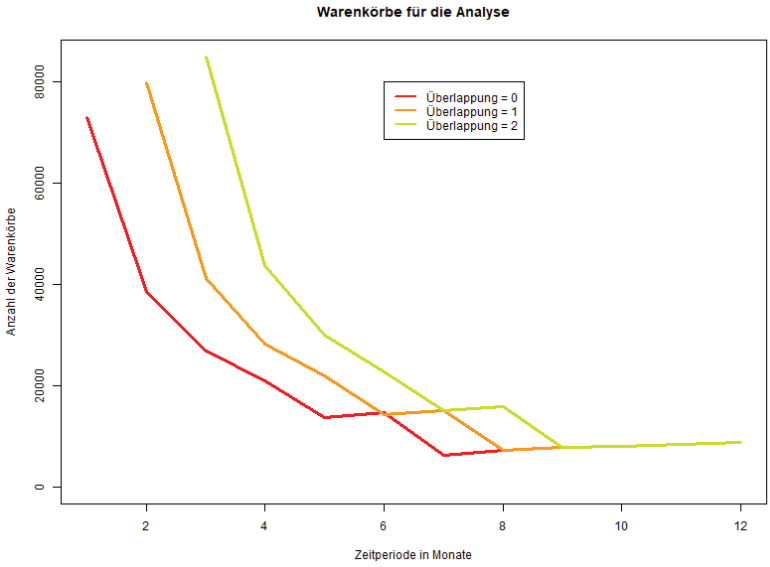

Abbildung 5.3: Ermittelte Warenkörbe für Testdatensatz B

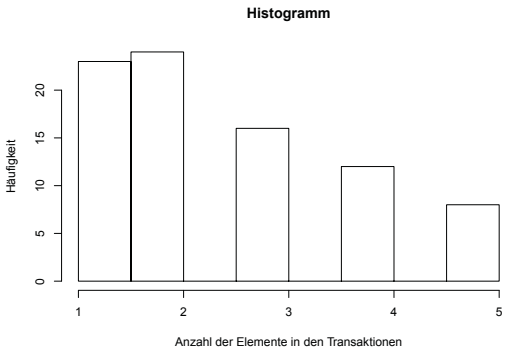

Abbildung 5.4: Warenkorbitems ($l = 12$)

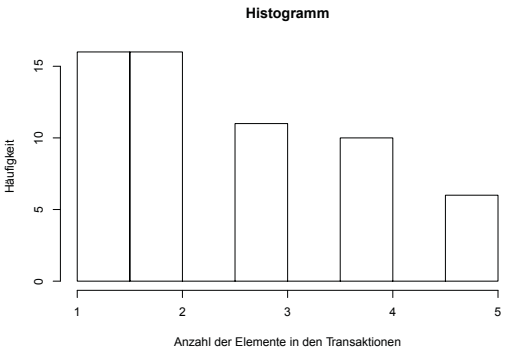

Abbildung 5.5: Warenkorbitems ($l = 18$)

die Häufigkeiten der gemeinsamen Nutzungen für die Buchpaare als Ähnlichkeitsmaß genutzt. Insbesondere sollen in der hierarchischen Darstellung eine Unterscheidung von Gruppen erfolgen, die eine hohe Unähnlichkeit aufweisen. Wie in beiden Abbildungen deutlich wird, sind die Unähnlichkeiten nicht sehr groß, so dass an dieser Stelle einerseits festgestellt werden kann, dass die Items im gemeinsamen Zusammenhang zwar mit kleinerer Intervalllänge ebenfalls geringer sind, aber andererseits der Unterschied nicht gravierend ist.

In Anlehnung zu Abbildung 5.1 wird in Abbildung 5.18 eine analoge Darstellung genutzt, um den Einfluss der Überlappung aufzuzeigen.

Mit einer geringeren Intervalllänge l ergeben sich mehr Warenkörbe. Diese sind aufgrund der zeitlichen Verteilung in den Daten, z.B. durch stärkere Nutzung zu Prüfungszeiten, nicht gleichverteilt. Somit ergeben sich hinsichtlich der Faktoren nur Schätzungen, die durch obere Schranken zusätzlich eingeschränkt werden können. Jedoch wird insbesondere für den Datensatz mit einer Intervalllänge von $l = 12$ deutlich, dass der Einfluss der Überlappung multiplikativ sein kann. Aber für realistischere Szenarien, in denen das Intervall wesentlich größer ist als die Überlappung, ist ein gestuftes Warenkorbwachstum zu beobachten. Für den Fall $l = 36$ sind die Warenkörbe konstant,

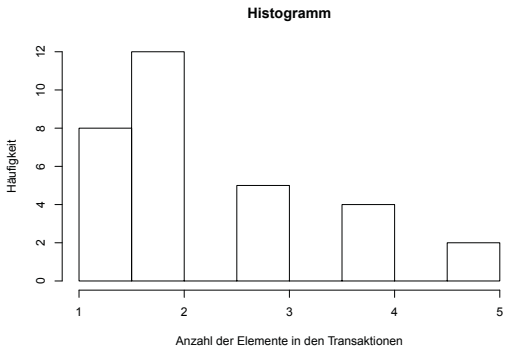

Abbildung 5.6: Warenkorbitems ($l = 24$)

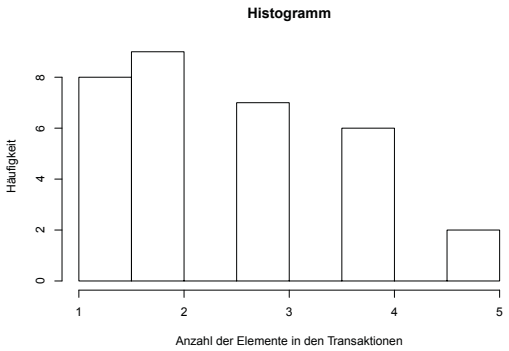

Abbildung 5.7: Warenkorbitems ($l = 30$)

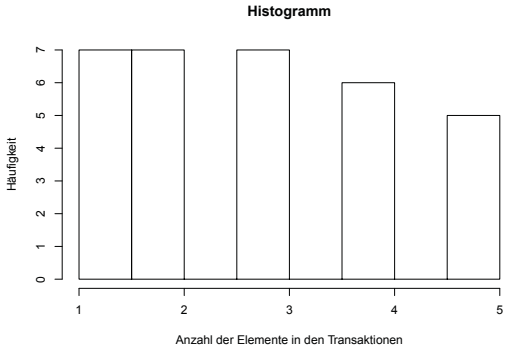

Abbildung 5.8: Warenkorbitems ($l = 36$)

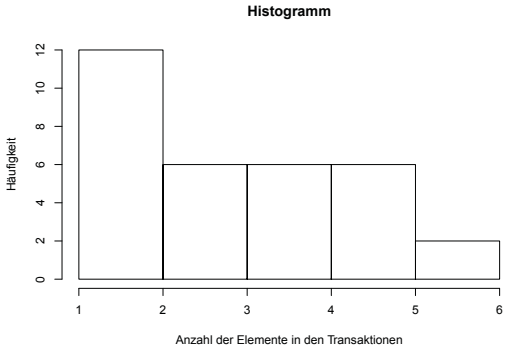

Abbildung 5.9: Warenkorbitems ($l = 44$)

da die Gesamtlaufzeit nicht ausreicht, um einen Einfluss zu sehen.

Der Faktor der Warenkörbe ist dahingehend wichtig, da mit ihm auch der Support höher gewählt werden sollte. Für die Intervalllänge von $l = 24$ wird deutlich, dass ein Plateau für die Überlappungen von 2 bis 8 vorliegt. In diesem Bereich erfolgt eine nähere Betrachtung der ermittelten Regeln in Abschnitt 5.3.2 für den Testdatensatz A.

5.3.2 Einfluss des Supports

Im folgenden Teil soll die Parametrisierung des Warenkorbverfahrens dargestellt werden. An dieser Stelle wird insbesondere der Support thematisiert.

Mit steigendem Support geht die Anzahl der Items genauso zurück wie die Anzahl der identifizierten Beziehungen. Während davon auszugehen ist, dass ein zu niedriger Support Bücher empfiehlt, die gar keine echte Relevanz aufweisen, führt ein hoher Support dazu, dass die Empfehlungsbasis sehr klein ist. Problematisch ist an dieser Stelle auch der Einfluss des im vorangegangenen Abschnitt beschriebenen Überlappungsgrades zu sehen. Denn in Abhängigkeit des Intervalls und der Überlappung kommen Empfehlungen mehrfach in die Analysebasis. Dies kann bis zum Überlappungswert auftreten. D.h. im Falle eines einmaligen Auftretens des Buches in der Ausgangsbasis, kann dieses Buch dann bis zu X-mal in der Warenkorbbasis vorkommen, wobei X dem Überlappungsgrad entspricht.

Prinzipiell wird in den Experimenten für den Testdatensatz A mit Supportwerten im Intervall $[1, 10]$ und für den Testdatensatz B im Intervall $[2, 10]$ in der Evaluation der Einfluss untersucht.

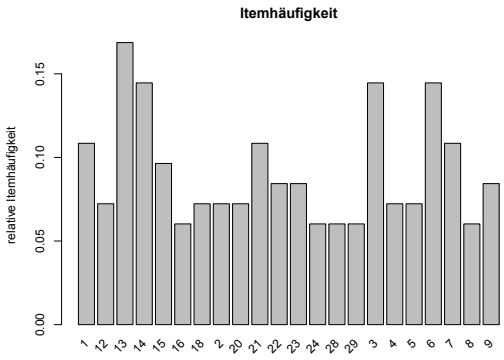

Abbildung 5.10: Itemfrequenzen ($l = 12$)

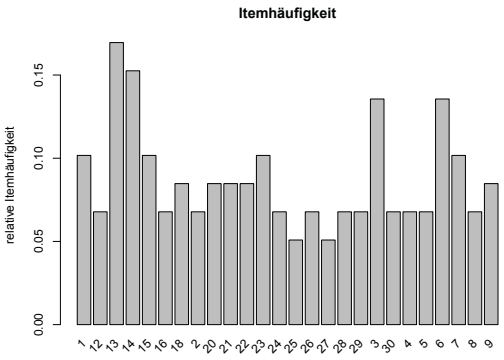

Abbildung 5.11: Itemfrequenzen ($l = 18$)

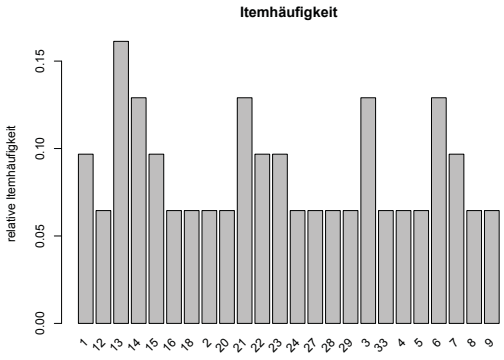

Abbildung 5.12: Itemfrequenzen ($l = 24$)

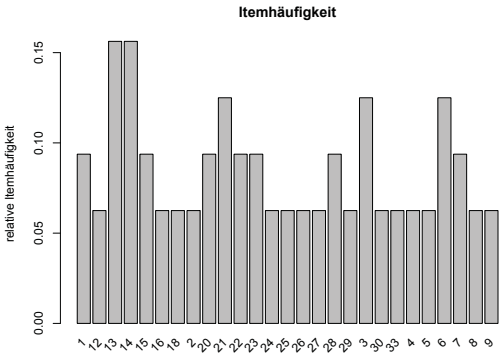

Abbildung 5.13: Itemfrequenzen ($l = 30$)

Theoretisch ist eine Betrachtung auch für das einmalige Auftreten als Grundlage für den Testdatensatz B möglich. Jedoch wird an dieser Stelle dann deutlich, dass die Systembeschränkungen ggfs. durch den Algorithmus ausgereizt werden. Während für die Implementierung des Apriori-Algorithmus für den Testdatensatz B bei einem Supportwert von 1 etwa 154 GB benötigt werden, reichen für alle größeren Supportwerte bereits 8 GB an Hauptspeicher aus. Für die Implementierung des FP-Growth-Algorithmus erfolgen die Konstruktionen des FP-Baumes, aber die Regelidentifikation scheitert bei kleinen Supportwerten am verfügbaren Heap-Speicher aufgrund der Rekursion des Algorithmus.

Daher soll an dieser Stelle kurz auch die Betrachtung erfolgen, welcher der beiden Algorithmen zu präferieren ist. Aufgrund des wesentlich größeren Rechenaufwandes und der damit besseren Kontrolle hinsichtlich Skalierung wird nur die Laufzeitbetrachtung für den Testdatensatz B durchgeführt. Für Testdatensatz A sind die benötigten Laufzeiten für die Berechnung der Regeln nicht valide ermittelbar, da sie sehr schnell erfolgt (in wenigen Millisekunden). Eine Analyse der Laufzeiten soll hier nur sehr kurz dargestellt werden und die Dimensionierung aufzeigen.

Einen direkten Vergleich des ursprünglichen Apriori-Verfahrens nach [10] wird in [119] durchgeführt. Problematisch ist an dieser Stelle die Tatsache, dass es bereits durch die Verbesserungen nach Borgelt und Kruse, durch den Einsatz von Baumstrukturen, eine Verbesserung für den Apriori-Algorithmus existiert [32]. [31] zeigt ebenfalls einen Implementierungsvergleich zwischen Apriori und dem FP-Growth-Algorithmus. Die in beiden Fällen beschriebenen Testfälle werden unter wesentlich größeren Supportwerten durchgeführt. Beide Studien kommen zu dem Ergebnis, dass

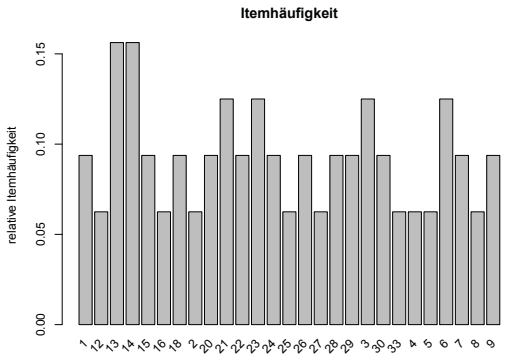

Abbildung 5.14: Itemfrequenzen ($l = 36$)

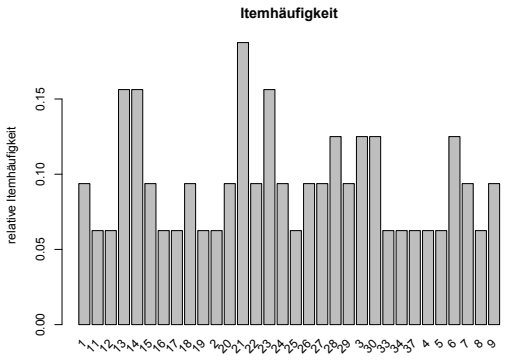

Abbildung 5.15: Itemfrequenzen ($l = 44$)

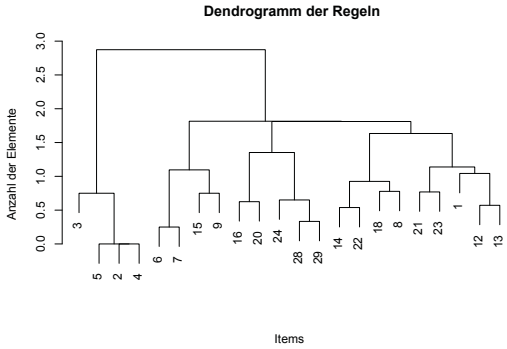

Abbildung 5.16: Regeldendrogramm ($l = 12$)

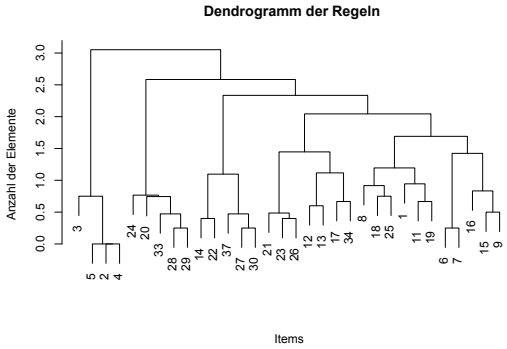

Abbildung 5.17: Regeldendrogramm ($l = 44$)

der FP-Growth-Algorithmus den Apriori-Algorithmus hinsichtlich der Performanz schlägt. Daher soll an dieser Stelle eine kurze Betrachtung der Laufzeiten für sehr geringe Supportwerte erfolgen. Dies ist insbesondere der Tatsache geschuldet, dass einerseits viele Bücher im Datensatz vorliegen, wobei jedoch aufgrund des Ausleihzeitraumes und des Bibliotheksangebotes nur wenige Ausleihen vorliegen.

Aufgrund der Systemumgebung für das Bibliothekssystem lassen sich beide Implementierungen nicht direkt nutzen. Daher erfolgt ein Vergleich beider Ansätze an dieser Stelle mit den folgenden aktuellen Implementierungen.

Die Apriori-Implementierung erfolgt im System R [127] mit dem arules Paket [76, 77]. Die Implementierung für den FP-Growth-Algorithmus ist an die SMILE-Umgebung[1] angelehnt, wurde jedoch für den Vergleich auf den FP-Growth-Algorithmus reduziert und für das Zielsystem in der aktuellen Java-Version kompiliert. Beide Umgebungen lassen sich im Kontext des Bibliothekssystems nutzen, so dass ein Datentransfer nicht notwendig ist. Dies bedeutet zugleich einen wesentlichen Vorteil für die Datenverarbeitung im Kontext der Data Privacy. Problematisch ist jedoch die Tatsache, dass die Ausführungen in unterschiedlichen Programmumgebungen erfolgen. In einem Benchmark wurde festgestellt, dass die SMILE-Umgebung die R-Implementierung für die binäre Klassifikation schlägt[2].

Für die Präsentation der Ergebnisse zur Laufzeit wird im Folgenden der Testdatensatz B gewählt

[1] SMILE: Statistical Machine Intelligence and Learning Engine, http://haifengl.github.io/smile/.
[2] Siehe hierzu den minimalen Benchmark für Machine Learning: https://github.com/szilard/benchm-ml.

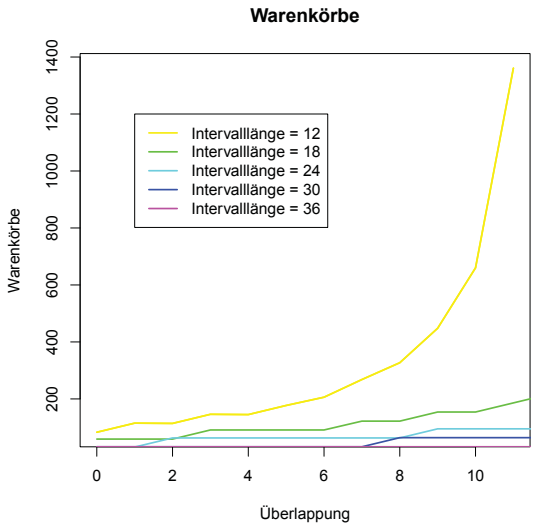

Abbildung 5.18: Warenkorbanzahl bei unterschiedlichen Überlappungen

mit dem Parameter 2 für die Überlappung. Bei den anderen Parameterwerten für die Überlappung ergibt sich ein analoges Bild, so dass an dieser Stelle der besseren Übersichtlichkeit auf die Darstellung des gesamten Parameterraums verzichtet wird.

In Abbildung 5.19 sind die Laufzeiten für beide Algorithmen mit den Supportwerten von zwei bis zehn abgebildet. Für die Intervalllänge der Warenkörbe liefern die Werte neun bis zwölf für beide Algorithmen eine ähnliche Laufzeit. Die beiden Werte liegen nah beieinander und mit geringerem Support erhöht sich die Laufzeit beider Algorithmen im exponentiellen Verlauf. Der Wert für den FP-Growth mit einer Intervalllänge von zwölf und dem Supportwert von zehn scheint ein Ausreißer zu sein, der ggfs. durch weitere Prozesse im Betriebssystem erklärt werden kann. Eine Wiederholung aller Experimente würde an dieser Stelle sicherlich zu einer Glättung führen.

Es wird ersichtlich, dass für den Supportwert zwei der Apriori-Algorithmus ein knapp schlechteres Laufzeitverhalten als der entsprechende FP-Growth Algorithmus aufweist. Mit steigendem Support ist dieser dann etwas schneller. Aufgrund der sehr kurzen Laufzeiten können beide Algorithmen als identisch performant angesehen werden. Dies steht in etwas anderem Kontext als die Evaluationen in der Literatur, da in der vorliegenden Arbeit insbesondere minimale Supportwerte gesetzt sind und somit eher eine Randanalyse des Laufzeitverhaltens an dieser Stelle erfolgt. Jedoch sind aufgrund der Datenlage und des Buchausleih-Empfehlungssystems diese Betrachtungen relevant. Beide Algorithmen liefern in allen Werten den gleichen Regelsatz.

Problematisch ist die Darstellung für die kleineren Intervalllängen. Hier ergeben sich mehr Warenkörbe bei zugleich anderen Belegungen der Items. Das bedeutet, die zu betrachtende Itemzahl erhöht sich stark. Für den FP-Growth ergeben sich hier bei kleinen Supportwerten Programmabbrüche, da der zur Verfügung stehende Heap-Speicher nicht ausreicht. Während der FP-Baum gebaut werden kann, wird aufgrund der Rekursion die Identifikation der Itemsets nicht erfolgreich durchgeführt. Einerseits erhöht sich die Laufzeit übermäßig, was nicht nur am exponentiellen Laufzeitverhalten des Algorithmus liegt, sondern auch der Tatsache des Java-Hauptspeichermanagements geschuldet ist, andererseits werden auch die Regeln nicht erstellt. Mit sinkender Intervalllänge nehmen die Programmabbrüche bei steigendem Support zu.

In Abbildung 5.20 wird daher nur der Apriori-Algorithmus mit seinen Laufzeiten dargestellt.

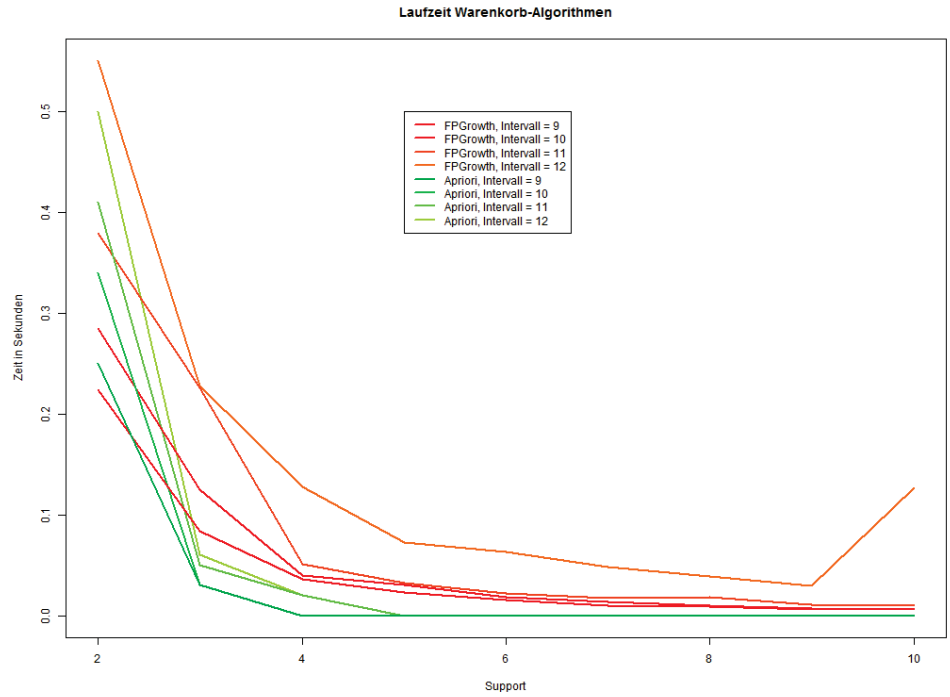

Abbildung 5.19: Laufzeiten der Algorithmen FP-Growth und Apriori für Testdatensatz B und Überlappung = 2

Das exponentielle Laufzeitverhalten wird weiterhin deutlich. Je geringer die Intervalllänge, desto höher muss der Support sein, um ähnliche Laufzeiten zu erzielen. Diese Verschiebung kann der Abbildung 5.20 ebenfalls entnommen werden. Für die Intervalllängen sieben und acht ergibt sich ein identisches Laufzeitverhalten. An dieser Stelle besteht der Warenkorb aus 15016 bzw. 15938 Transaktionen, wobei die jeweilige zu betrachtende Itemanzahl (in Abhängigkeit des Supports) ebenfalls ähnlich ist. Für die entsprechende Anzahl der Transaktionen wird auf Abbildung 5.3 verwiesen.

Bei der Laufzeitbetrachtung wurde das Abspeichern der Regeln nicht betrachtet, da dies sehr von dem HDD-Zugriff abhängig ist. Somit können sowohl für den FP-Growth- wie auch für den Apriori-Algorithmus ähnliche Laufzeiten identifiziert werden. Aufgrund der gleichen Ergebnismengen, solange kein Programmabsturz erfolgt, wird im weiteren Verlauf der Arbeit keine Unterscheidung zwischen beiden Algorithmen vorgenommen. Da der Apriori-Algorithmus stabil läuft, werden die Ergebnisse dieses Algorithmus' für alle weiteren Betrachtungen verwendet.

5.3.3 Supportanalyse für Testdatensatz A

Der Einfluss des Supports kann für den Testdatensatz neben der quantitativen Beschreibung auch in grafischer Form hinsichtlich der Güte der ermittelten Regeln erfolgen. Aufgrund der unterschiedlichen Einflussparameter wird hier zunächst eine Ceteris Paribus Analyse durchgeführt.

Aus dem Parameterraum wird im Folgenden der Parametersatz Länge des Intervalls 18 Monate, Überlappung 0 und 12 ausgewählt. Für die anderen Parameter ergeben sich ähnliche Ergebnisse.

In Abbildung 5.21 sind die Regeln nach Support und Lift dargestellt. Die Intensität der Farbe gibt zudem die Konfidenz an. Es wird hier ersichtlich, dass die Supportabstände sehr diskret sind, dies ist durch die geringe Anzahl an Items zu begründen. In Abbildung 5.22 wird deutlich, dass die

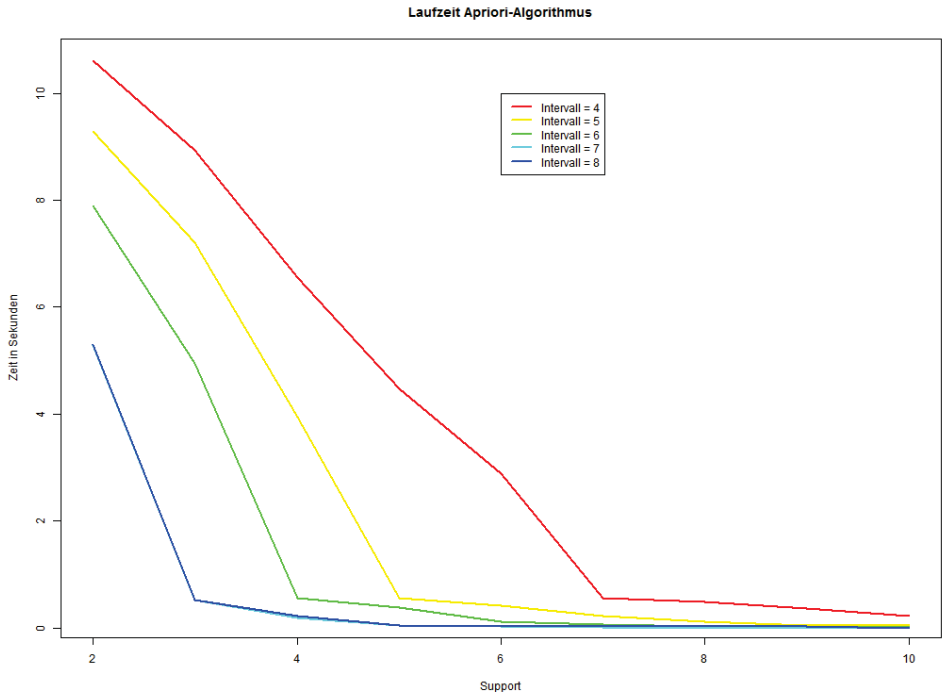

Abbildung 5.20: Laufzeiten den Apriori-Algorithmus bei kleinen Supportwerten für Testdatensatz B und Überlappung = 2

Supportabstände geringer sind. Auch ist der maximale Lift größer. Aber es bleibt fraglich, ob die richtigen Elemente miteinander in Beziehung stehen.

Dies soll eine weitere Darstellung klären. An dieser Stelle wird als Supportwert 2 für den Algorithmus festgesetzt. Dies entspricht einem Support von knapp einem Prozent. Es werden zwei von Hahsler und Karpienko vorgeschlagene Visualisierungen zunächst genutzt [74].

Basierend auf den Ergebnissen von [38, 62, 92, 120] wird ein Graph genutzt, um die Zusammenhänge zwischen den Büchern darzustellen. Aufgrund der geringen Anzahl an Regeln und Büchern ist es möglich, die vollständige Repräsentation zu wählen. Oftmals wird diese netzwerkbasierte Analyse nur mit den wichtigsten Regeln durchgeführt [74].

In Abbildung 5.23 wird der Netzwerkgraph für die Warenkörbe mit Zeitintervall 18 Monate und keiner Überlappung dargestellt. Zunächst werden drei Empfehlungsgruppen deutlich, die keinen gemeinsamen Bezug aufweisen. In der Darstellung sind zudem der Support entsprechend der Größen der Kreise und der Lift entsprechend der Farbintensität in den Kreisen dargestellt. Somit ist z.B. der höchste Lift bei den Buchempfehlungen für Nr.1 und Nr.10 zu beobachten. An dieser Stelle wurden die ordnungserhaltenen Wörterbuchcodierungen der Übersichtlichkeit wegen genutzt. Ein Nachschlagen im entsprechenden Wörterbuch ergibt, dass es sich bei Nr. 1 um die PPN 02408542 handelt und Nr. 10 entspricht der PPN 13306435. Bei der Identifikation der anderen Bücher und einer Gesamtbetrachtung ergibt sich, dass die drei Cluster im Wesentlichen aufgrund der Fachgebiete zuzuordnen sind.

Das bedeutet, aufgrund der Eingabedaten, die durch drei Gebiete bestimmt sind, wird bei einem Support von zwei auf der Titelebene deutlich, dass auch in den ermittelten Empfehlungen diese Clusterbildung vorhanden ist. Dies ist einerseits ein positives Ergebnis, da die thematischen Beziehungen so auch unmittelbar in die Empfehlungen einfließen können. Andererseits verleitet dieser

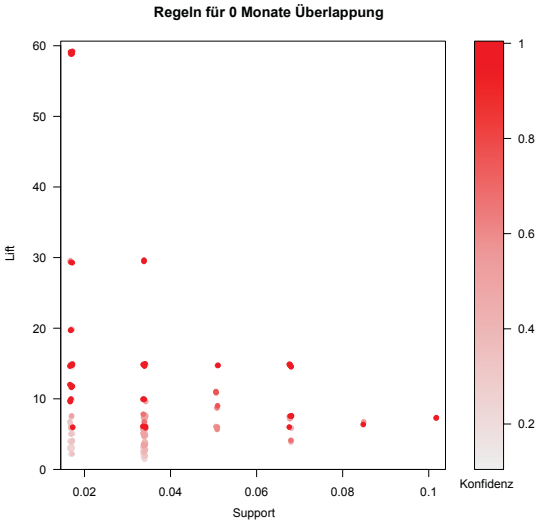

Abbildung 5.21: Scatterplot 18 Monate und keine Überlappung

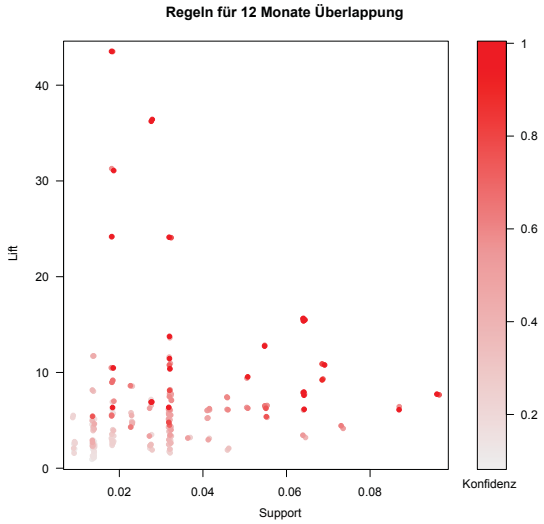

Abbildung 5.22: Scatterplot 18 Monate und 12 Monate Überlappung

Ansatz eventuell auch dazu, Bücher in das Themengebiet einzuordnen, was jedoch unzutreffend ist. Denn so verbirgt sich hinter Nr. 25 z.B. ein Buch ohne Zuordnung zum Themengebiet Ingenieurwesen. Auch im Bereich Informatik befinden sich Bücher ohne diese thematische Zuordnung.

Ein weiterer Aspekt in diesem Zusammenhang ist, dass nicht alle Bücher eine Empfehlungsregel aufweisen. Nur 32 der 38 Bücher sind in diesem Regelsatz enthalten. Für die anderen Bücher reicht die Datenbasis nicht aus.

Um den Zusammenhang zwischen Support und Regelsatz zu verdeutlichen, sind in Abbildung 5.24 und Abbildung 5.25 analog die netzwerkbasierten Darstellungen erfolgt — an dieser Stelle einerseits für den Supportwert von eins und andererseits für den Wert von vier.

Aus den beiden Abbildungen wird ersichtlich, dass bereits in Abbildung 5.24 alle Informationen für Abbildung 5.25 vorhanden sind. Wie zu erwarten, nimmt die Anzahl der Empfehlungen mit steigendem Support ab. Die Kreisgröße deutet aber auch auf die Verknüpfungen, die weiterhin im Netzwerk bestehen bleiben, da der Support das Kriterium ist, nach dem die Filterung erfolgt.

Zusätzlich wird hier deutlich, dass bei zu kleinem Support die Empfehlungsmenge ansteigt. Jedoch sind nicht alle Empfehlungen gleichwertig, sondern die Cluster sind meist durch einzelne Knoten (z.B. Nr. 21 oder Nr. 25) verbunden. Zu beachten ist außerdem, dass die Farbintensität in den Grafiken unterschiedlich skaliert ist. Der Lift nimmt mit steigendem minimalen Support ab.

Eine zweite Darstellungsweise für den Zusammenhang und die leichtere Interpretation der Assoziationsregeln ist die gruppierte Matrixdarstellung [74], wie in Abbildung 5.26 präsentiert. Hierbei können wiederum nur bedingt viele Elemente dargestellt werden. Für diese Art der Visualisierung bietet sich noch der Testdatensatz A an. Die Assoziationsregeln werden in diesem Verfahren geclustert. Dabei kommt die Jaccard Distanz, wie von Gupta et al. vorgeschlagen, zum Einsatz [72]:

$$d_{Jaccard}(X_i, X_j) = 1 - \frac{|X_i \cap X_j|}{|X_i \cup X_j|}.$$

In der Abbildung 5.26 werden sowohl die Köpfe wie auch die Rümpfe der Regeln in Beziehung gesetzt. Dabei erfolgt zudem eine Sortierung entsprechend des Lifts, so dass Regeln mit großem Lift im oberen Teil der Matrix erscheinen. Der Lift selbst wird durch die Farbintensität wiedergegeben.

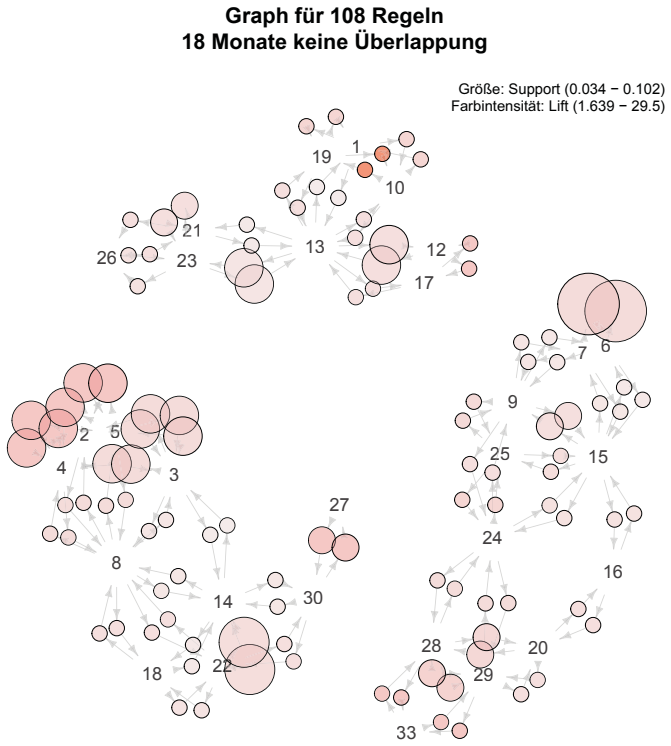

Abbildung 5.23: Graphdarstellung Regeln 18 Monate keine Überlappung

Auch der Support wird grafisch repräsentiert, indem die entsprechende Kreisgröße genutzt wird. Für die Abbildung wurde der minimale Supportwert von zwei gewählt, daher erfolgt bereits hier eine Filterung und nur 30 Bücher werden durch 108 Regeln dargestellt. Mit steigendem minimalen Support reduziert sich die Darstellung entsprechend.

5.3.4 Supportanalyse für Testdatensatz B

An dieser Stelle soll eine kurze Übersicht erfolgen, welchen Einfluss sowohl auf die Regelmenge als auch auf die sich ergebenden Items der Support für den Testdatensatz B hat.

Zunächst werden in Abbildung 5.27 die Itemanzahlen angezeigt, für die eine Empfehlung vorliegt. Mit steigendem Support muss die Empfehlungsmenge sinken. Für den Supportwert von zwei ergeben sich bei der Intervalllänge $l = 1$ die meisten Elemente. Dies liegt darin begründet, dass anscheinend viele Bücher mehrfach durch Nutzer ausgeliehen werden und somit bei kürzeren Intervalllängen häufiger Berücksichtigung finden. Da keine Überlappung vorliegt, kann eine mehrfache Ausleihe durch denselben Nutzer in diesem Fall zu einer Empfehlung führen. Zu beachten ist an dieser Stelle, dass Verlängerungen nicht in den Ausleihdaten abgebildet werden. Dies bedeutet auch, dass um eine möglichst große Menge an Empfehlungen zu generieren, die Intervalllänge nicht den Gesamtzeitraum umfassen sollte. Zwischen den Intervalllängen $l = 2$ und $l = 4$ ist eine lineare Abnahme zu verzeichnen. Der Knick bei $l = 5$ für den Supportwert zwei ergibt sich aus der Tatsache, dass einige Elemente nicht ausreichend gut zusammengeführt werden, bei $l = 6$ wird aufgrund der Gesamtbetrachtung des Zeitraums wieder ein besseres Ergebnis erzielt.

Für die Betrachtung der Überlappung ergeben sich ähnliche Konstellationen, wobei der zu vermutende Anstieg der Empfehlungsmenge durch das mehrfache Einfließen in die Warenkörbe zu erklären ist. An dieser Stelle soll insbesondere der Abdeckungsgrad der empfohlenen Items

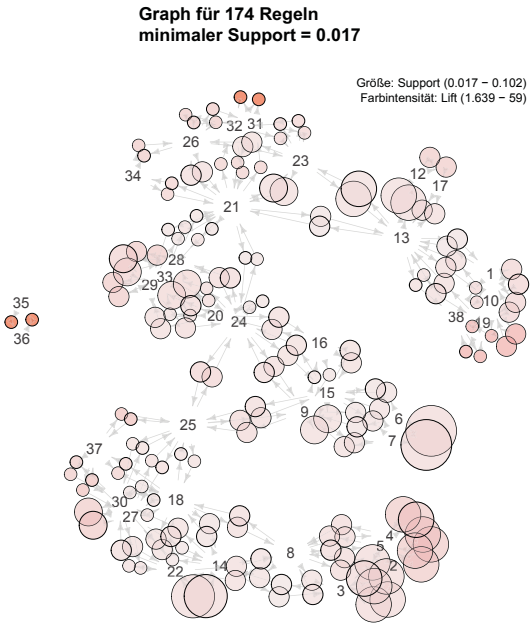

Abbildung 5.24: Graphdarstellung für 18-0 mit Supportwert = 1

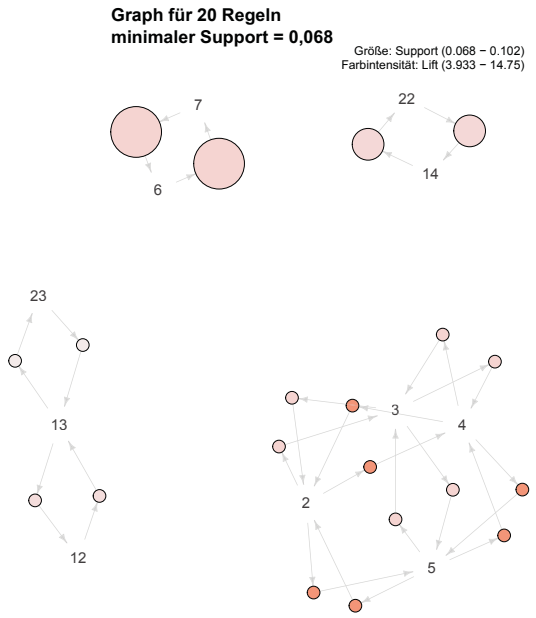

Abbildung 5.25: Graphdarstellung für 18-0 mit Supportwert = 4

betrachtet werden. Dieser ergibt sich als Quotient aus den empfohlenen Items und den Items der Datenbasis.

In den Abbildungen 5.28 und 5.29 wird deutlich, dass mit einem steigendem Überlappungsgrad der Intervalle auch die Empfehlungsmenge ansteigt. Die Kurvenverläufe sind ähnlich, nur die leichte Rechtsverschiebung und eine Steigerung in den Kurven ist zu sehen. Dies kann aufgrund der Parametrisierung erwartet werden.

5.3.5 Trimming der Regeln

Um eine Differential Privacy zu gewährleisten, werden nicht alle Empfehlungen in die Datenbank übernommen, sondern nur die mit dem größten Lift. Wie im vorangegangenen Kapitel beschrieben, wird die maximale Menge der Empfehlungen auf zehn festgesetzt. Daher soll an dieser Stelle geprüft werden, welchen Einfluss diese Reduzierung auf die Empfehlungen hat.

Anhand des Überlappungsgrades von drei für den Testdatensatz A wird dies für zwei Supportwerte verdeutlicht. In den Abbildungen 5.30 und 5.31 sind für das Intervall 18 die Größe der Regeln angegeben. Bei einem Supportwert von eins in Abbildung 5.30 wird bereits deutlich, dass nur zwei Regelsets davon betroffen sind. Somit werden insgesamt etwa fünf Prozent der Regeln eingeschränkt. Zudem wird deutlich, dass mit zunehmendem Support wesentlich weniger Regeln betroffen sind. Es kann festgestellt werden, dass eine Einschränkung durch Weglassen von Regeln zu eher unbedeutenden Informationsverlusten führt. Hierbei ist zusätzlich zu berücksichtigen, dass wichtige Regeln (definiert durch den Lift) beibehalten werden.

Letztlich soll auch noch am Testdatensatz B eine exemplarische Überprüfung des Trimmings für den Datenbestand für das Gesamtintervall ohne Überlappung bei einem Supportwert von zwei erfolgen.

Auch an dieser Stelle wird deutlich, dass nur ein sehr geringer Teil der Regeln vom Weglassen der Elemente betroffen ist. Die Mehrzahl der Regeln bleibt unberührt und durch die Verschleierung

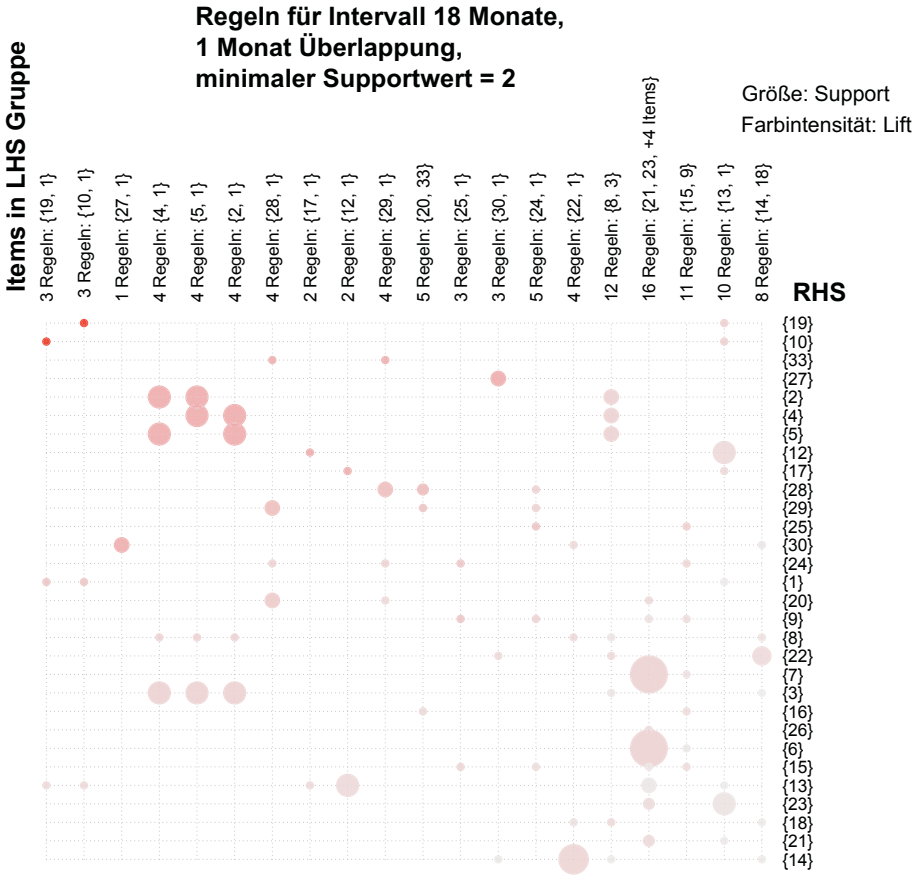

Abbildung 5.26: Gruppierte Matrixdarstellung der Regeln für $l = 18$ und $overlap = 1$

der Informationen werden insbesondere Bücher entfernt, die sich durch wenige Nutzer mit hohen Ausleihzahlen (Heavy User) ergeben. Dies liegt daran, dass bei steigendem Support diese eher entfallen.

Der Anteil der reduzierten Regeln zum originalen Regelsatz ist in Abbildung 5.34 für Testdatensatz A mit einer Überlappung von sechs und in Abbildung 5.35 für Testdatensatz B mit keiner Überlappung aufgezeigt.

Mit steigendem Support nimmt die Anzahl der herausgelöschten Regeln ab, da nur wenige Regeln mehr als zehn Items enthalten. Für den Testdatensatz A muss ein hoher Überlappungsgrad gewählt werden, so dass eine Löschung erfolgt. Diese erfolgt dann auch nur für sehr wenige Regeln. Am Testdatensatz B wird deutlich, dass die Intervalllänge einen großen Einfluss auf die gekürzte Regelmenge hat. Jedoch muss an dieser Stelle ebenfalls festgehalten werden, dass für größere Intervalllängen bei kleinem Support der Anteil der gekürzten Regeln sehr klein ist.

5.4 Einfluss der Hierarchie

Der Einfluss der Hierarchie wurde in der Literatur bereits ausführlich diskutiert und im Ergebnis empfohlen, diese Elemente bei der Warenkorbanalyse zu verwenden, die dadurch verbessert wird. An dieser Stelle soll jedoch ein anderer, eher nutzerzentrierter Ansatz aufgegriffen werden — mit diesem

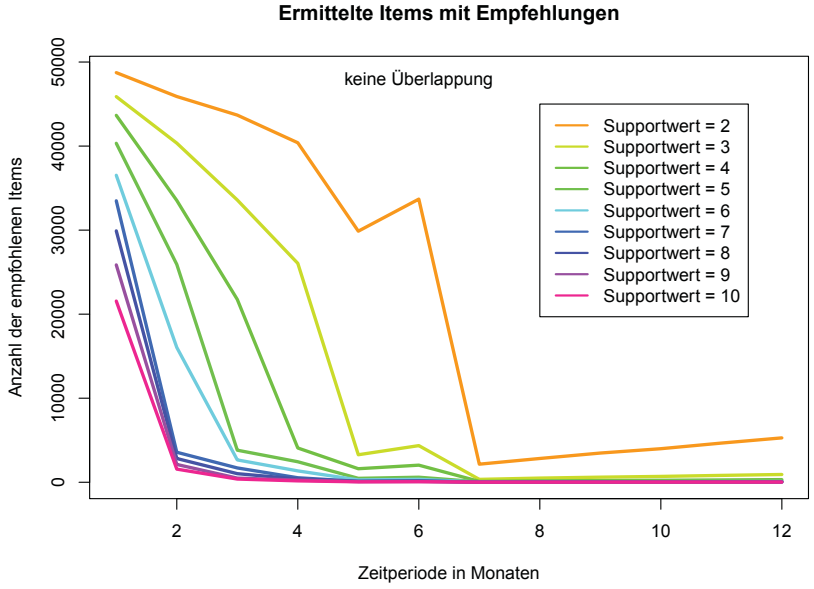

Abbildung 5.27: Ermittelte Items für Warenkörbe ohne Überlappung

Hierarchiewechsel ergibt sich ein anderer Fokus. So kann es für den Fachreferenten von großem Interesse sein, welche Exemplare besonders häufig genutzt bzw. nicht-genutzt werden. Hierzu wird er jedoch aufgrund der Vielzahl der Bücher keine Navigation im OPAC durchführen, sondern sich anhand von Listen einen Überblick verschaffen. Trotzdem ist es an dieser Stelle wichtig, auch die Warenkorbanalyse heranzuziehen, denn dadurch können insbesondere Gemeinsamkeiten identifiziert werden, die für die Buchaufstellung genutzt werden können.

Aus Sicht der OPAC-Nutzer ist eine Empfehlung auch dann hilfreich, wenn diese nicht nur für eine bestimmte Auflage vorliegt, sondern beispielsweise auch ältere Auflagen empfohlen werden, sofern die Neuere zu diesem Zietpunkt nicht zur Verfügung steht. Aber auch die Empfehlung auf Autorenebene scheint vielversprechend zu sein, um zumindest ähnliche Arbeiten zu finden. Problematisch an dieser Stelle ist jedoch, dass in den Katalogen keine einheitlichen Verzeichnungen hinsichtlich der Auflagen oder Autoren erfolgen. Daher wird zwar anhand des Testdatensatzes A dieses Element kurz dargestellt, jedoch ist zu bedenken, dass für eine automatisierte Lösung, weitere Anforderungen notwendig sind, um z.B. die Datenintegration mittels GND der DNB zu nutzen. Somit stellt dieser Ansatz ein Proof-of-Concept dar. Die weitere Gestaltung muss in zukünftigen Arbeiten erfolgen.

5.4.1 Empfehlungen auf Basis EPN

Eine Empfehlung auf EPN-Basis macht für den Nutzer wenig Sinn, da er zwischen den einzelnen Exemplaren nicht unterscheidet. Jedoch kann es für Fachreferenten wichtig sein, Bücher zu identifizieren, die auf Ebene der Exemplare relativ häufig ausgeliehen werden. Es kann nicht nur erkannt werden, ob Ersatz für ein bestimmtes Buch beschafft werden muss, sondern auch, an welchen Stellen eine Bestandsoptmierung notwendig ist.

Für eine direkte Verwendung der EPN im Kontext der Warenkorbanalyse ist eine Anpassung der Abfrage im LBS-System notwendig, denn ein Rückschluss auf die EPN von der PPN ist nicht möglich. An dieser Stelle wird im Testdatensatz A (T-A) daher nur die Abfrage aus Abschnitt 4.1.1

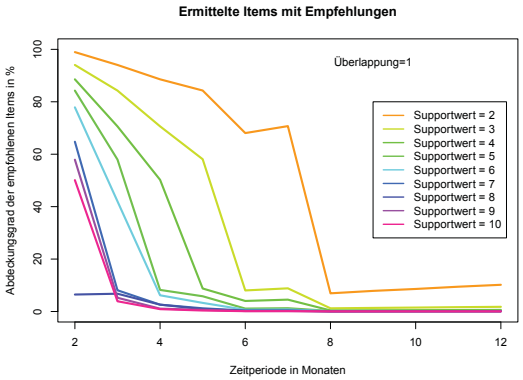

Abbildung 5.28: Itemabdeckung Überlappung=1

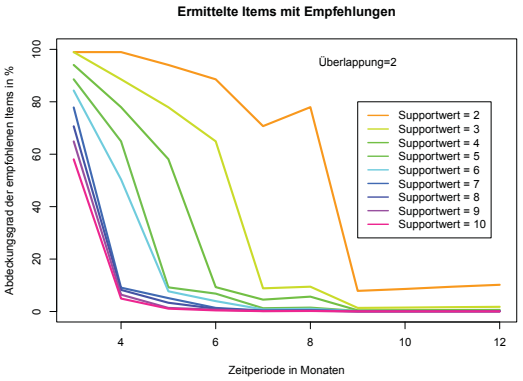

Abbildung 5.29: Itemabdeckung Überlappung=2

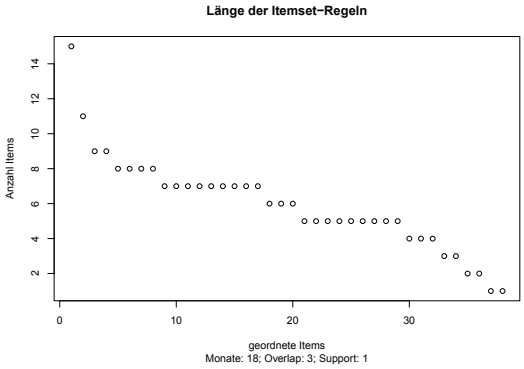

Abbildung 5.30: Anzahl der Regelelemente Support=1

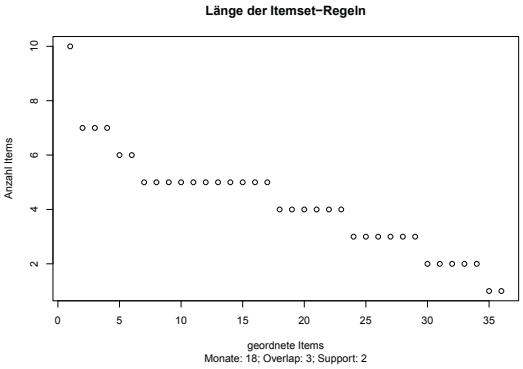

Abbildung 5.31: Anzahl der Regelelemente Support=2

hinsichtlich der EPN umgewandelt werden. Für den Testdatensatz A führt dies nun zu den folgenden Ergebnissen.

In den Abbildungen 5.36 und 5.37 sind die identifizierten Regeln für die Intervalllängen $l = 12$ und $l = 18$ abgebildet.

Hierbei wird deutlich, dass mit höherer Überlappung die maximale Regelanzahl beibehalten wird. Zudem werden die Regeln mit zunehmender Intervalllänge reduziert. Für den Support schrumpft erwartungsgemäß die identifizierte Regelbasis mit zunehmenden Support. Im Vergleich zur Basis PPN werden weniger Regeln identifiziert. Um den Vergleich weiter zu vertiefen, werden zusätzlich noch die empfohlenen Bücher in den Abbildungen 5.38 und 5.39 abgebildet.

Während auf PPN-Basis 32 Titel vorliegen, sind auf EPN-Basis 38 Bücher vorhanden. Für kurze Intervalllängen werden diese bei großer Überlappung auch analog zu der PPN komplett in der Regelmenge behalten. Mit geringem Überlappungsgrad nehmen die Empfehlungen etwas stärker ab. Für die Überlappungen 0 bis 2 und 3 bis 6 (in Abbildung 5.39 sehr deutlich zu sehen) sind die Kurven sehr ähnlich. Daher ist der Einflussbereich des Intervalls in bestimmten Bereichen weniger wichtig.

Somit kann festgehalten werden, dass, obwohl die Menge der Items und damit verbunden die Menge der Regeln leicht ansteigt, eine Warenkorbanalyse auch auf dem Niveau EPN möglich ist.

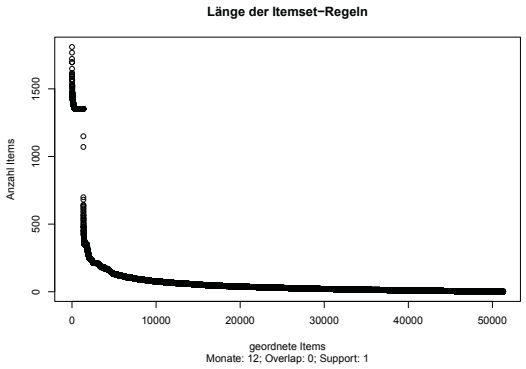

Abbildung 5.32: Anzahl der Regelelemente Testdaten B-1

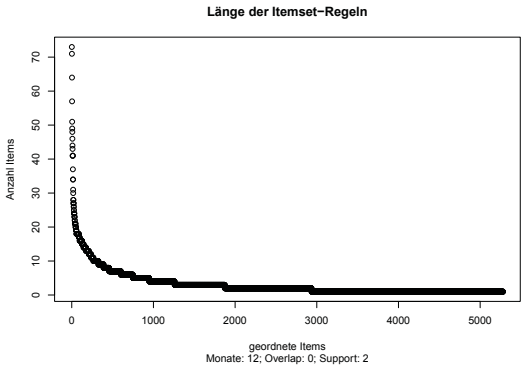

Abbildung 5.33: Anzahl der Regelelemente Testdaten B-2

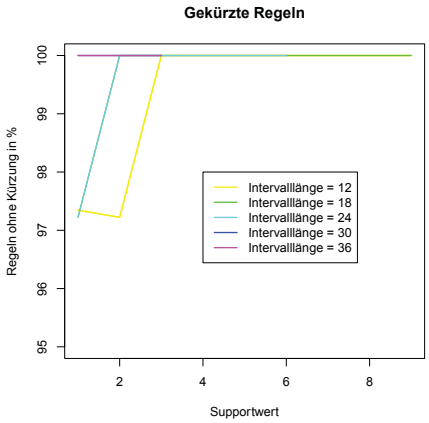

Abbildung 5.34: Testdatensatz A, Überlappung=6

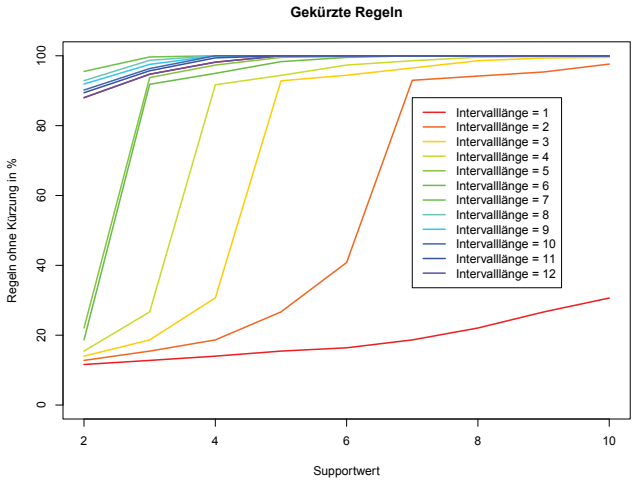

Abbildung 5.35: Pruning der Regelelemente Testdaten B, Überlappung = 0

5.4.2 Empfehlungen auf Basis des Titels

Die Empfehlungen auf Basis des Titels können durch ein einfaches Mapping der EPN oder PPN erfolgen. Da hier eine direktionale Abhängigkeit vorhanden ist, lässt sich dies beispielsweise direkt in die transaktionalen Warenkörbe einbetten. Somit sind im Procedere bis auf das Mapping keine Änderungen notwendig. Dies kann in analoger Weise zur EPN-Verwendung erfolgen.

An dieser Stelle soll ebenfalls eine kurze Betrachtung zu der Regelmenge und den Empfehlungen analog zum vorangegangenen Abschnitt durchgeführt werden.

In den Abbildungen 5.40 und 5.41 sind die identifizierten Regeln abgebildet für die Intervalllängen $l = 12$ und $l = 24$.

Auch hier wird wiederum deutlich, dass mit kleinerer Intervalllänge mehr Regeln identifiziert werden. Wie anzunehmen, sinkt jedoch die Gesamtregelmenge, da weniger Items (nur noch 25) zur Verfügung stehen. Der Übersichtlichkeit halber wurden nur noch die Überlappungsgrade von 0, 2 und 6 als Repräsentanten ausgewählt. Wie zuvor ergibt sich für längere Intervalllängen eine größere Einheitlichkeit bei den Überlappungen. So ist in Abbildung 5.41 leicht identifizierbar, dass es fast keinen Unterschied zwischen den Regeln von keiner Überlappung und einer Überlappung von 2 gibt.

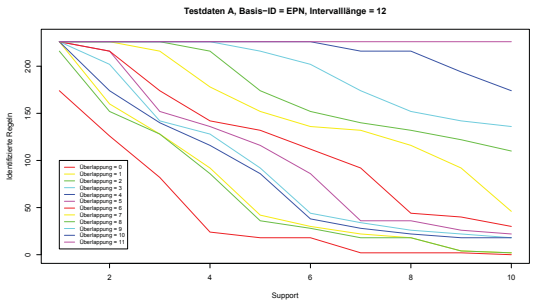

Abbildung 5.36: Regeln T-A, EPN, $l = 12$

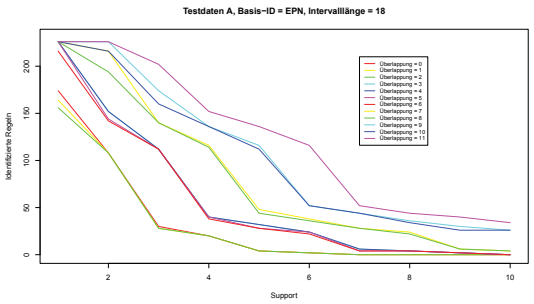

Abbildung 5.37: Regeln T-A, EPN, $l = 18$

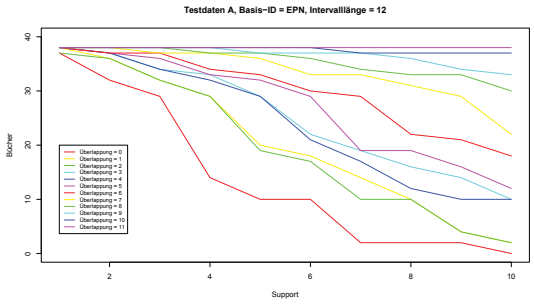

Abbildung 5.38: Bücher T-A, EPN, $l = 12$

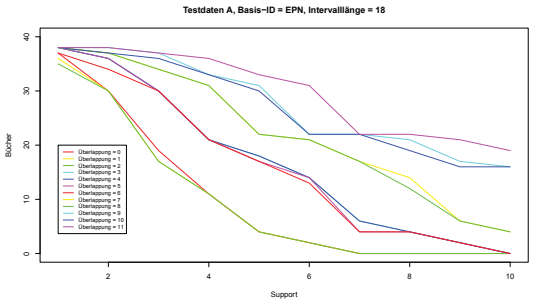

Abbildung 5.39: Bücher T-A, EPN, $l = 18$

Für kleinere Intervalllängen ist zudem deutlich, dass die Abweichung geringer ausfällt.

In den Abbildungen 5.42 und 5.43 sind zudem die enthaltenen Titelvorschläge abgebildet.

Diese sind wiederum ähnlich zu den vorangegangenen Betrachtungen. Somit lässt sich auch hier das Fazit ziehen, dass eine Empfehlung auf Titelebene zwar eine verringerte Anzahl der Elemente bedeutet, aber prinzipiell möglich ist.

5.4.3 Empfehlungen auf Basis der Autoren

Für die Empfehlung auf Autorenbasis ergibt sich die Herausforderung, dass die Warenkörbe entweder als Einzeltransaktion, d.h. der Form TransaktionsID, ItemID vorhanden sein müssen oder im Warenkorb durch Integerwerte als Ersatz für die Identifzierer repräsentiert werden. Da für einzelne Bücher mehrere Autoren vorhanden sein können, muss daher eine Transformation gefunden werden, die diesen Aspekt inkludiert. Zunächst kann dies erfolgen, indem im Integer-Warenkorb der EPN oder PPN eine Ersetzung erfolgt für die Erstautoren inklusive Markierung, falls mehrere Autoren beteiligt sind. Daran anschließend können die weiteren Autoren der Mapping-Datei hinzugefügt und schließlich die Marken ersetzt werden. Alternativ müsste für jeden weiteren Autor eine Transaktion hinzugefügt werden und somit sich die transaktionalen Warenkörbe erweitern. Für das Proof-of-Concept wird an dieser Stelle der erste Weg genutzt.

Für den Testdatensatz A existieren 58 Autoren. Daher nimmt bei gleicher Warenkorbmenge die Regelmenge zu, denn die Warenkörbe enthalten nun mehr Items.

In den Abbildungen 5.44 und 5.45 sind die ermittelten Regelmengen wieder abgebildet. Während man für die Intervalllänge $l = 12$ nun eindeutig mehr Regeln sieht, ergeben sich auch für die Intervalllänge $l = 24$ mehr Regeln. Jedoch sind diese mit steigendem Support stark abfallend.

Letztlich soll noch einmal anhand der Graphendarstellung aufgezeigt werden, wie die Autorenemp-

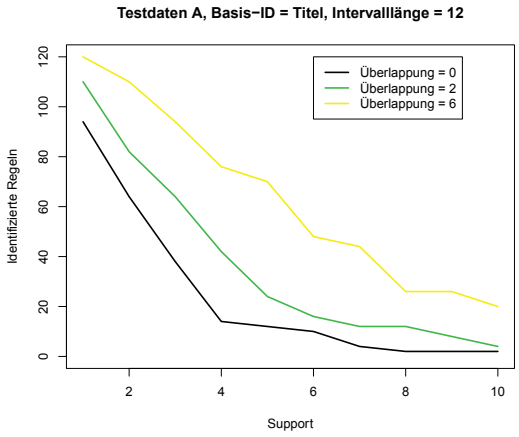

Abbildung 5.40: Regeln T-A, Titel, $l = 12$

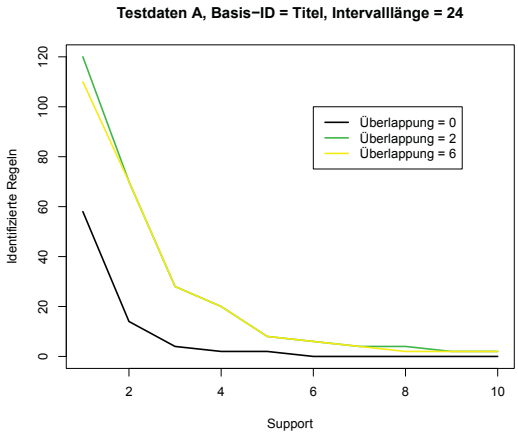

Abbildung 5.41: Regeln T-A, Titel, $l = 24$

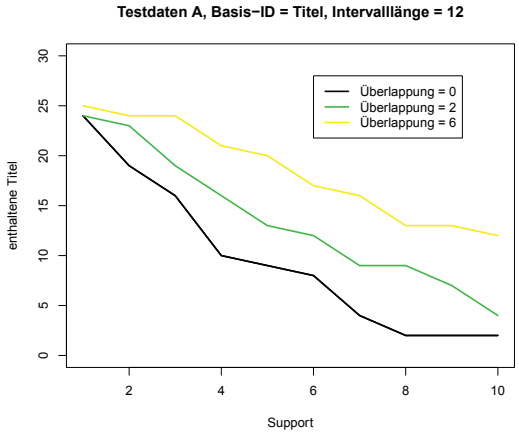

Abbildung 5.42: Bücher T-A, Titel, $l = 12$

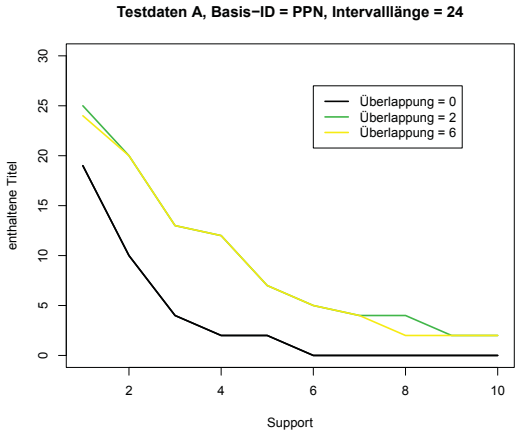

Abbildung 5.43: Bücher, T-A, Titel, $l = 24$

fehlungen zusammenhängen. In Abbildung 5.46 ist für das Intervall $l = 12$ ohne Überlappung bei einem Supportwert von 3 die Regelmenge dargestellt. In dieser Abbildung wurden die Autorennamen direkt in der Visualisierung genutzt. In Abbildung 5.47 erfolgt die Darstellung für $l = 24$ ohne Überlappung mit einem Supportwert von 2.

Wie in Abbildung 5.46 erkennbar sind die bereits im vorangegangenen Teil identifizierten Cluster wieder sichtbar: Psychologie (links unten), Informatik (links oben), den Bereich Ingenieurwesen (großer Cluster rechts unten) sowie einen Vertreter aus dem nicht zugeordneten Bereich. Für steigende Supportwerte werden die Empfehlungen geringer und somit die Cluster kleiner. Mit einem größeren Intervall nehmen die Cluster ebenfalls ab. Obwohl ein geringerer Supportwert (2) genutzt wird, ist die Regelmenge wesentlich kleiner. Trotzdem lassen sich auch hier einige Zusammenhänge noch ablesen. Zusammenfassend kann auch bestätigt werden, dass es möglich ist, eine höhere Hierarchieebene in der Warenkorbanalyse zu nutzen. Für Autoren ist dies mit einem etwas höheren Aufwand verbunden, da mehr als ein Autor an einem Werk beteiligt sein kann. Dies bedeutet, dass es notwendig ist, den Prozess bereits in der Konzeptionsphase zu berücksichtigen. Andernfalls müssen weitere Transformationen durchgeführt werden, auf die an dieser Stelle nicht weiter eingegangen werden soll.

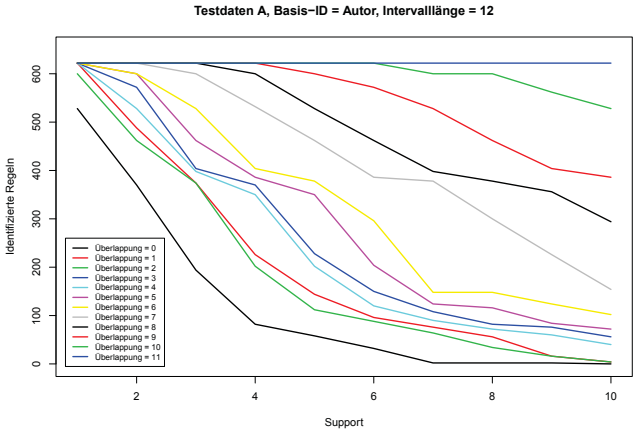

Abbildung 5.44: Testdatensatz A Autoren, $l = 12$

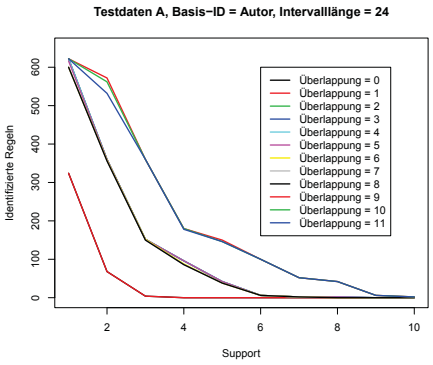

Abbildung 5.45: Testdatensatz A Autoren, $l = 24$

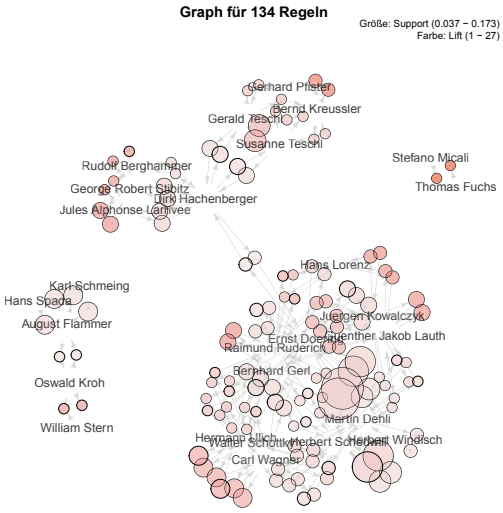

Abbildung 5.46: Graphendarstellung für Autorendaten, $l = 12, s = 3$

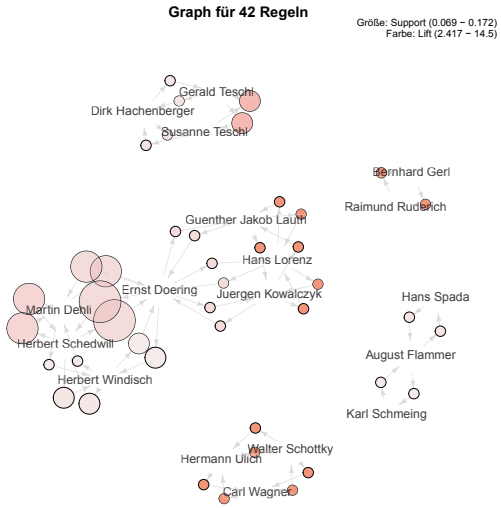

Abbildung 5.47: Graphendarstellung für Autorendaten, $l = 24, s = 2$

5.5 Einfluss des Vergessens

Ein Vergessen erfolgt im Kontext der Warenkorbanalyse in zwei unterschiedlichen Zusammenhängen. Zunächst ist die Aufteilung in Intervalle eine Möglichkeit, Zusammenhänge aufzulösen und somit nicht nur mehr Transaktionswarenkörbe zu erzielen, sondern damit einhergehend frühere Ausleihen nicht in den selben Korb zu legen. Dadurch wird der sich über die Zeit ergebene Wechsel in der Nutzerintention abgebildet. Für wissenschaftliche Bibliotheken kann es beispielsweise Sinn machen, die Intervalllänge auf ein bis vier Semester einzugrenzen, da einerseits die verfügbare Literatur wechseln kann und sich andererseits die aufgrund der Heterogenität der Studiengänge vorhandene Vielfalt besser abgrenzen lässt.

Um nahe bzw. wesentliche Zusammenhänge nicht zu verlieren, bietet sich die Verwendung eines Überlappungsbereiches an. Auch dieser wurde im vorangegangenen Abschnitt bereits betrachtet. Wichtig ist an dieser Stelle, dass die Überlappung adäquat gewählt wird. Eine Justierung dieses

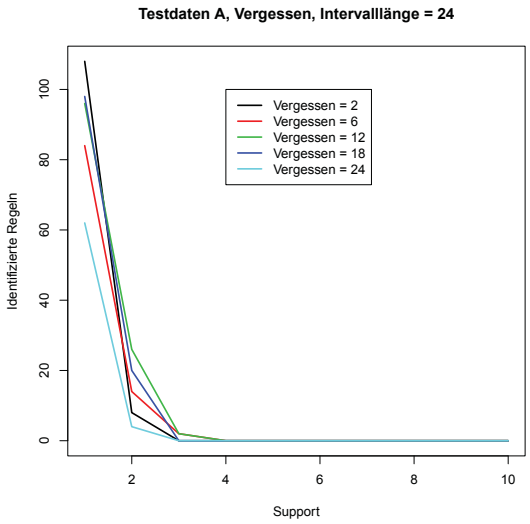

Abbildung 5.48: Identifzierte Regeln, $l = 24$ mit Vergessen

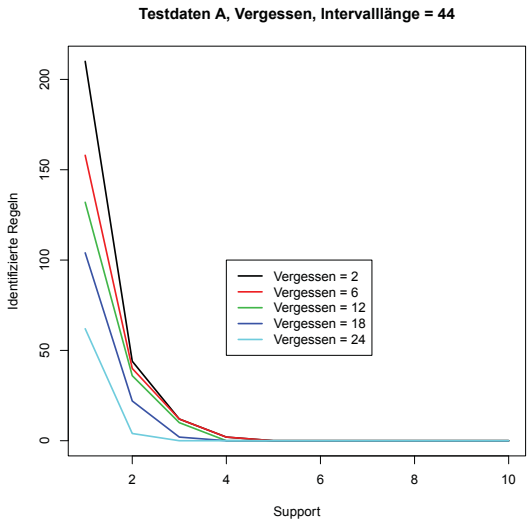

Abbildung 5.49: Identifzierte Regeln, $l = 44$ mit Vergessen

Parameters erfordert hinreichend viele Beobachtungen, die sowohl im Testdatensatz A wie auch B (aufgrund des kurzen Intervalls) nicht ausreichend sind. Eine diesbezügliche erste Vermutung ist, dass sich Werte in Abhängigkeit der Intervalllänge ergeben, diese aber prinzipiell wesentlich kleiner sein sollten als das Intervall.

Eine weitere Betrachtung soll an dieser Stelle noch mittels dem Vergessen der Ausleihdaten erfolgen. Dabei können diese durch Löschen der entsprechenden Daten nicht mehr für die Warenkorbanalyse zur Verfügung stehen. Aufgrund der systemgegebenen Verhältnisse sind diese Daten nicht mehr im LBS vorhanden und können somit auch nicht mehr reproduziert werden.

Im Testdatensatz A ergeben sich 44 Ausleihintervalle. Das Löschen der ältesten Daten erfolgt für die Intervalle von eins bis 24. Dabei werden die Warenkörbe und Regeln betrachtet, die sich dann noch ergeben. Die Hierarchieebene ist für alle Betrachtungen in diesem Fall die PPN. Auch hinsichtlich der Intervalllänge werden nur zwei Repräsentanten gewählt, der volle Bereich mit Überlappung von 0 und der Bereich von 24 mit Überlappung von einem Intervall.

Vergessen bezieht sich an dieser Stelle auf die Transaktionen in einem Zeitraum. Für die Evaluation werden die ersten 2, 6, 12 und 24 Monate als Zeiträume gewählt, die nicht in die Datenbasis mehr einbezogen werden. In den Abbildungen 5.48 und 5.49 sind die Regeln abgebildet, die sich nach dem Vergessen noch in Abhängigkeit des Supports ermitteln lassen.

Dabei lässt sich einerseits sehen, dass kleinere „Vergessensbereiche“ nur einen geringen Einfluss auf die Regelmenge haben. Für kürzere Intervalllängen ist der Einfluss größer. Während der Gesamtbereich kaum eine Änderung in der Regelmenge für ein Vergessen der ersten 12 Monate ausweist (z.B. im Supportwert 2 oder 3), bedeutet ein Vergessen im Testdatensatz A für die Intervalllänge von 24 mit einem Monat Überlappung bereits drastische Änderungen, denn für den Supportwert von 3 gibt es keine Regeln mehr. Für die Darstellung der Änderung der Regeln soll nun noch exemplarisch die Graphendarstellung genutzt werden. Hierbei wird nur der Gesamtbereich bei einem Supportwert von 2 genutzt.

In den Abbildungen 5.50 bis 5.53 wird die Abnahme der Empfehlungen deutlich, die mit dem Vergessen einhergeht. Aufgrund der Unterschiede durch den Algorithmus im Mapping ist ein direkter Vergleich nicht möglich, sondern es müssen die jeweiligen Mappings betrachtet werden. Dabei wird

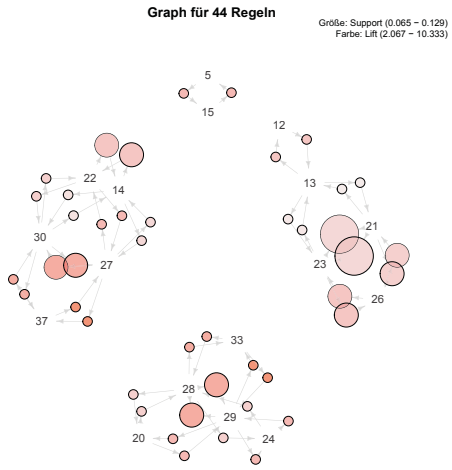

Abbildung 5.50: Graphendarstellung Vergessen = 2

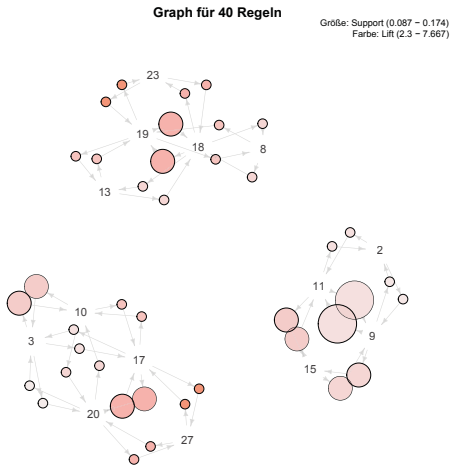

Abbildung 5.51: Graphendarstellung Vergessen = 6

deutlich, dass die Cluster in ihrer Zuordnung zum Fachgebiet erhalten bleiben, auch wenn die Elemente sich reduzieren.

Zusammenfassend lässt sich für das Vergessen festhalten, dass für längere Intervalle ein Vergessen eher unproblematisch ist, so dass eine stabile Empfehlungsbasis auch dann noch ausgesprochen werden kann, wenn beispielsweise 25 Prozent der Daten „Vergessen“ werden. Dies bedeutet hinsichtlich der Data Privacy aber auch im Kontext der Nutzermodellierung, aufgrund seiner sich über die Zeit ändernden Präferenzen, ein sinnvolles Mittel für die Warenkorbanalyse. In der Praxis sind jedoch größere Zeiträume zunächst aufzunehmen, bevor eine Analyse hinsichtlich der Robustheit erfolgen kann.

5.6 Evaluation Testdatensatz B im Praxiseinsatz

Für die letzte Evaluation werden die Daten des anschließenden Monats März 2018 aus dem LBS der UB Magdeburg genutzt. Hierbei wurden 7005 Transaktionen registriert. Dabei werden nur die Ausleihen betrachtet. Eigentlich sollte für eine Analyse direkt das Präsentationssystem OPAC genutzt werden, um die dort entstehenden Klicks auszuwerten. Jedoch ist aufgrund der Implementierungskomplexität und des Systembetriebs diese Integration nicht erfolgt. Daher wird auf die Ausleihebene fokussiert.

Zunächst ist dabei zu beachten, dass im wissenschaftlichen Ausleihbetrieb auch häufig Bücher ausgeliehen werden, die neu beschafft wurden. Insbesondere bei Erwerbungsvorschlägen ist dies der Fall. Daher wird in einem ersten Schritt ermittelt, welche Bücher nicht bereits im Datenbestand von Testdatensatz B zur Verfügung stehen und diese werden für die weitere Betrachtung herausgefiltert. Dabei handelt es sich um insgesamt 3938 verbleibende Bücher auf Titelebene.

Für den Abdeckungsgrad wird der Testdatensatz B mit einem Überlappungsintervall von 1 Monat genutzt. Sowohl die Intervalllänge l als auch der Support haben einen Einfluss auf den Abdeckungsgrad. In Abbildung 5.54 ist der Support mit dem entsprechenden Abdeckungsgrad aufgezeigt.

Dabei wird deutlich, dass die Kurven mit steigendem Support stark sinken. Trotzdem ist insbesondere für die kurze Intervalllänge auch bei einem Supportwert von 10 noch über 65 Prozent der

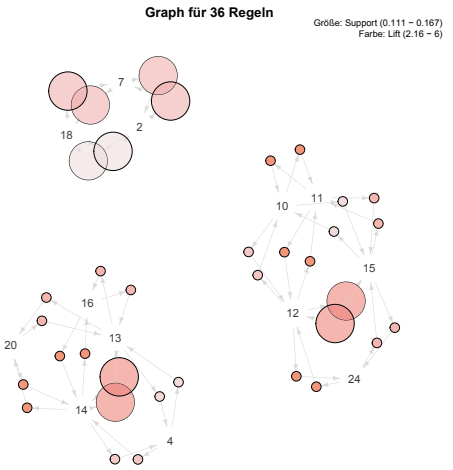

Abbildung 5.52: Graphendarstellung Vergessen = 12

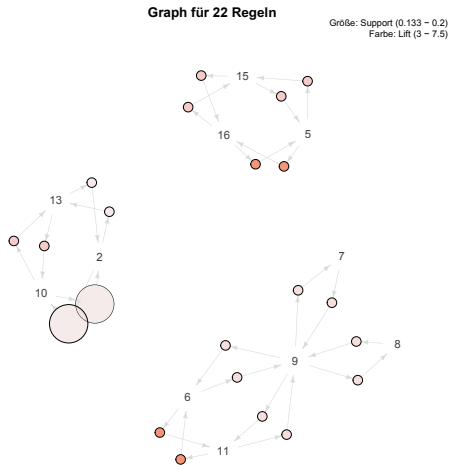

Abbildung 5.53: Graphendarstellung Vergessen = 18

ausgeliehenen Titel im Folgemonat Teil der Empfehlungsbasis. Bei insgesamt 25971 Titeln in der Empfehlungsbasis ($l = 2, s = 10$) werden noch 2587 der 3938 empfohlen. Für einen Supportwert von 2 sind von den 51280 Titeln in der Empfehlungsbasis 3918 im Testsatz enthalten. Somit kann eine Abdeckungsrate an dieser Stelle von mehr als 99 Prozent erzielt werden.

Im Vergleich dazu sind für den Gesamtzeitraum ($l = 12$) wesentlich weniger Elemente in der Empfehlungsbasis und folglich auch weniger Treffer für den Testsatz. Für den Supportwert von 2 sind gerade einmal 5280 Titel vorhanden und aus dem Testsatz können nur 921 empfohlen werden. Die Intervalllängen 12 bis 7 unterscheiden sich nicht merklich. In Abbildung 5.55 sind zum besseren Vergleich noch die Intervalllängen zum Abdeckungsgrad abgebildet.

Die bereits getroffenen Aussagen können auch hier nachvollzogen werden. Aufgrund des gewählten Überlappungsgrades von 1 lässt sich der beschränkte Anstieg für $l = 7$ in allen Kurven erklären. Mit geringem Supportwerten und Intervalllängen lassen sich sehr hohe Abdeckungsgrade ermitteln. Aber auch ein Anstieg der Intervalllänge kann mit reduziertem Support teilweise aufgefangen werden.

Somit kann resümiert werden, dass die Warenkorbanalyse, wenn auch die Ursprungsdaten noch ein recht kleines Zeitfenster aufweisen bereits sehr hohe Empfehlungsergebnisse erzielt werden können. Titel, die neu in den Bibliotheksbestand aufgenommen wurden, werden dabei aber explizit nicht betrachtet. Auch selten genutzte Titel können nicht immer ausreichend berücksichtigt werden. Trotzdem ist die gewählte Lösung in der Lage, für einen Großteil der Ausleihvorgänge des Folgemonats bereits Empfehlungen anzubieten.

5.7 Bewertung der Forschungsfragen

Der Abschnitt fasst kurz die wichtigsten Punkte hinsichtlich der Forschungsfragen zusammen. Dabei wird auf die Ergebnisse der Evaluation aber auch die in Kapitel 4 gesetzten Maßnahmen kurz eingegangen.

Forschungsfrage 1 Wie können Data-Privacy-Anforderungen für ein Buchausleih-Empfehlungssystem umgesetzt werden?

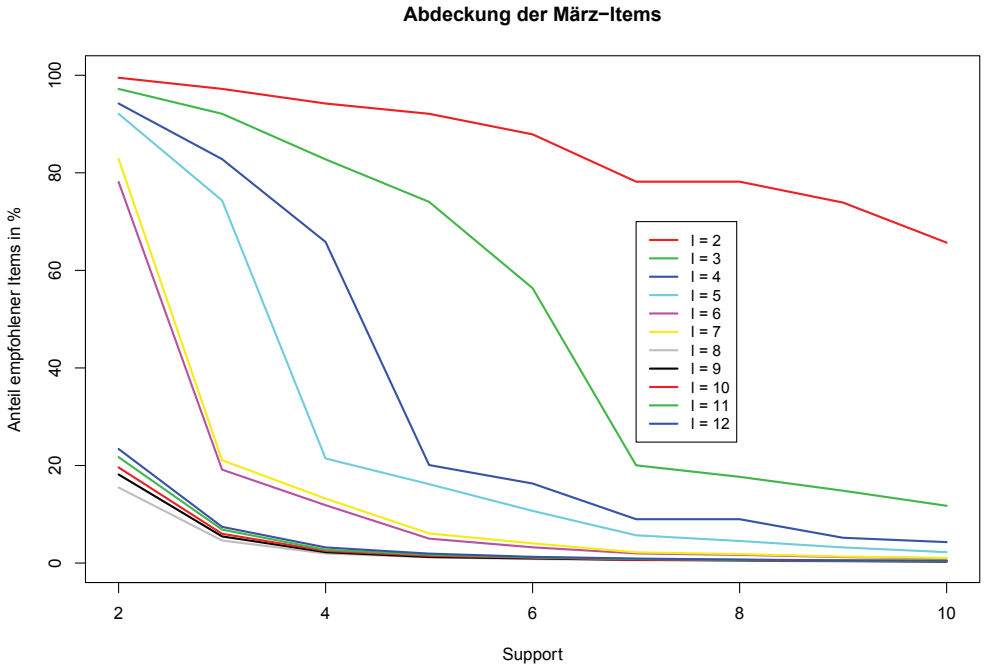

Abbildung 5.54: Abdeckungsgrad der Empfehlungsmenge für Ausleihen im Folgemonat

Personenbezogene Daten werden durch unterschiedliche Gesetze aber auch Nutzerinteressen geschützt. Dabei muss beachtet werden, dass diese mit der neuen Datenschutzgrundversorgung dem Nutzer in jeglicher Weiterverwendung bekannt sind und ggfs. eine Zustimmung eingeholt werden muss. Daher sind Data-Privacy-by-Design Ansätze in diesem Zusammenhang notwendig. Eine einfache Anonymisierung reicht nicht aus, wenn sich aus den Daten Rückschlüsse auf die Personen ziehen lassen. Daher ist der in der Warenkorbanalyse genutzte Parameter Support zwar ein Ansatz, dieser kann jedoch nur in beschränktem Maße vergleichbar der k-Anonymität angesehen werden.

Jedoch ist es mittels gestuften Verfahren von der Anonymisierung der Identifizierer bis hin zur randomisierten und beschränkten Ausgabe der Ergebnisse möglich, ein Konzept zu verfolgen, dass ein hohes Maß an Data Privacy ermöglicht. Durch das Herauslösen des Personenkontextes kann daher auch für das Ausleihsystem einer wissenschaftlichen Bibliothek ein Empfehlungssystem gestaltet werden. Dabei sind die Ergebnisse im längeren Kontext gültig, als beispielsweise Browsersessions, die einen wesentlich kleineren Zeitraum betrachten. Die Ergebnisse benötigen einerseits eine breite Basis, jedoch können auch Ansätze aus dem Bereich der Data Privacy genutzt werden, um den Datenbereitstellungszeitraum für das Empfehlungssystem gesetzten Anforderungen gerecht zu werden.

Somit ist das in Kapitel 4 beschriebene Vorgehen tauglich, um Data Privacy Anforderungen umzusetzen und zugleich ein Empfehlungssystem zur Verfügung zu stellen, dass fachspezifische Empfehlungen geben kann. Eine Personalisierung und somit maßgeschneiderte Empfehlungen für den Nutzer sind dabei explizit nicht vorgesehen.

Forschungsfrage 2 Wie lassen sich hierarchische Strukturen für die Warenkorbanalyse sinnvoll einsetzen?

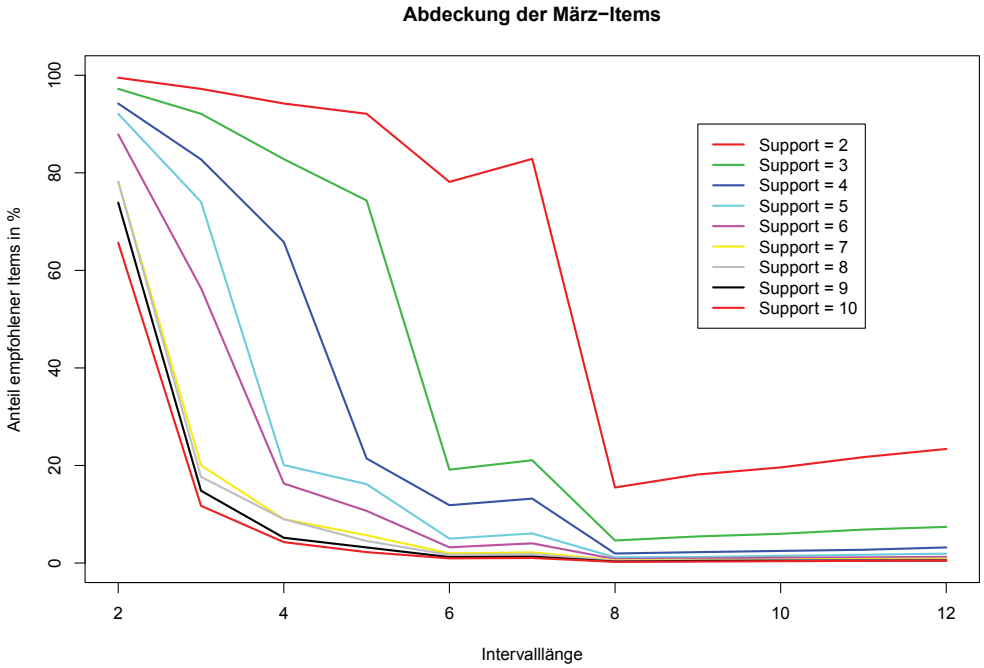

Abbildung 5.55: Abdeckungsgrad der Empfehlungsmenge im Folgemonat (Intervalllänge)

Die im bibliothekarischen Umfeld genutzten Hierarchien lassen sich auch dahingehend nutzen, dass diese im Empfehlungssystem berücksichtigt werden können. So kann der Kontext geändert werden, z.B. von der Titelebene hin zum Autor, ohne das inhärente Zusammenhänge aufgelöst werden. Auch eine Verfeinerung zur Ebene der Exemplare ist möglich, bedient dann aber eine andere Nutzergruppe, z.B. Fachreferenten für den Kontext des Bestandsmanagements. Mit der kleineren Anzahl an Elementen auf höherer Hierarchie-Ebene gehen ggf. verbesserte Empfehlungen hinsichtlich des Bewertungskontextes durch einen höheren Support einher. So können dem Nutzer weitere Empfehlungen angeboten werden. Offen bleibt hierbei aber die Frage, ob im wissenschaftlichen Kontext Altauflagen, die bei der Werkebene ebenfalls inkludiert werden, hilfreich sind, oder aufgrund der sich häufig ändernden und neuen wissenschaftlichen Publikationen dem Nutzer ungeeignet erscheinen.

Forschungsfrage 3 Welche Strategie zum Vergessen älterer Ausleihen führt zu ausreichenden Empfehlungen?

Das Vergessen kann in zwei unterschiedlichen Kontexten erfolgen. Zum einen können ältere Transaktionsdaten einfach unberücksichtigt bleiben. Negative Auswirkungen ergeben sich dann, wenn die Empfehlungsbasis recht klein ist. Der Einfluss ist allerdings gering, wenn die Empfehlungsdatenbasis umfangreich ist.

Ein weiterer erfolgsversprechender Ansatz ist in diesen Szenarien ein Vergessen der Ausleihgeschichte der Nutzer durch Aufteilung in kleinere Zeitabschnitte durchzuführen. So erhöht sich die Empfehlungsdatenbasis, sodass umfangreiche Empfehlungen generiert werden können. Diese zweite Strategie des Vergessens führt einerseits dazu, dass über einen längeren Zeitraum gemeinsam ausgeliehene Bücher nicht mehr gemeinsam identifiziert werden. Trotzdem ergeben sich aufgrund

der Aufteilung hinreichend mehr Empfehlungen im System. Für wissenschaftliche Bibliotheken mögen Zeitintervalle auf Basis von ein bis zwei Semestern hinreichend sinnvoll sein. Somit kann auch für diese Forschungsfrage eine Identifikation bestehend aus einem Mix von Vergessen der Beziehungen der Items und Vergessen von Transaktionen wie in Kapitel 4 und 5 aufgezeigt werden.

6 Fazit

Einerseits spielt Datenschutz eine immer höhere Rolle im Umgang mit automatisierten Verfahren. Andererseits sind Techniken notwendig, um mit der Flut an Informationen und Angeboten umzugehen. Die Suche nach der sprichwörtlichen Nadel im Heuhaufen gestaltet sich auch in wissenschaftlichen Bibliotheken schwierig: Welches ist das beste Buch für meine Studien, wenn die angegebene Pflichtliteratur nicht ausreicht? Die Suche nach Arbeiten kann beispielsweise am Regal erfolgen, so die Aufstellungssystematik dies unterstützt. Jedoch werden diese Regalbesuche seltener und sind häufig zeitraubend.

Heutzutage sind Empfehlungssysteme in Webseiten so eingebettet, dass Nutzer damit wie selbstverständlich agieren und diese nutzen. Daher ist es nur konsequent, diese auch im wissenschaftlich-bibliothekarischen Kontext anzubieten. Jedoch muss bei der Lösungsimplementierung darauf geachtet werden, keine spezielle Lösung für ein System zu erstellen, sondern vielmehr alle Suchsysteme wie OPAC und Discovery-System zu berücksichtigen. Somit kann der zentrale Datenplatz zum Ausleihverkehr als Datenbasis genutzt werden, um unter Beachtung der Data Privacy Anforderungen auch Empfehlungen zu Büchern zu geben, die von anderen Nutzern ausgeliehen wurden.

6.1 Zusammenfassung

Die vorliegende Arbeit konzentriert sich auf die Entwicklung und Evaluation eines prototypischen Systems für Empfehlungen basierend auf den Ausleihverkehrsdaten. Diese sind zwar nicht personengebunden, jedoch ist es notwendig, gemeinsam ausgeliehene Bücher zusammenzufassen. Dabei sind dann ggfs. Identifikationen der Nutzer möglich. Im Sinne der Data Privacy sollten die Nutzerdaten nur nach Zustimmung verarbeitet werden oder eben gänzlich unberücksichtigt bleiben. Da eine Erlaubnis auch jederzeit zurückgezogen werden kann und zugleich die für die Warenkorbanalyse weiteren notwendigen Attribute in die Datenhaltung einfließen müssten, ist dieser Ansatz nicht realisierbar.

Daher wurde ein Empfehlungssystem konzipiert, dass Data-Privacy-By-Design verfolgt, um so einen Einsatz in unterschiedlichen Systemen für die wissenschaftliche Literatursuche zu gewährleisten. Durch die gewachsene Komplexität von bibliothekarischen Anwendungen sind dabei neben den Bestimmungen zum Datenschutz und den Anforderungen an die Data Privacy eine Vielzahl von Methoden, Systemen und Werkzeugen zusammenzubringen, die die effiziente nachhaltige Nutzung des Systems ermöglichen.

Ein wichtiger Aspekt ist die Integration der Lösung in die bibliothekarische Systemlandschaft. So wird zugleich sichergestellt, dass die Datensicherheitsanforderungen berücksichtigt werden. Dies schränkt jedoch Lösungen ein, die nur außerhalb der Plattform genutzt werden können. Dabei kommen sowohl auf Entwicklungsumgebung die Sprachen Java, R und PHP zum Einsatz, wie auch auf Datenhaltungsebene relationale Datenbanken und Dateien.

In der Verbundanalyse werden sehr unterschiedliche Verfahren seit Jahrzehnten erforscht und eingesetzt. Mit der Zunahme der Datenmengen bieten sich Data-Mining-Verfahren an. Für die Verbundanalyse hat sich der Apriori-Algorithmus als einfaches, aber mächtiges Instrument herausgestellt.

Aufgrund der Kandidatengenerierung ist die Effizienz jedoch nicht gegeben. Dabei können jedoch die Weiterentwicklungen des Apriori-Algorithmus nach [31] und der FP-Growth-Algorithmus [79] genutzt werden. Wie in Kapitel 5 gezeigt, bietet die erste Variante stabilere Ergebnisse für kleine Supports. In der Literatur weist der FP-Growth-Algorithmus eine bessere Performanz auf, was aber in der Tatsache der kleinen Supports liegen kann. Auch die gewählten Implementierungen sind ggfs. nicht kompetitiv.

Darüber hinaus ist es komplex, die Parametrisierung durchzuführen, angefangen von der Warenkorbdefinition bestehend aus Intervalllänge und Überlappung, über die Support- und Konfidenzniveaus bis hin zur Auswahl, wie viele Empfehlungen präsentiert und wie viele in der Datenbasis bereitgehalten werden. Während einige Konstellationen kaum Änderungen bringen, sind andere Einflüsse bedeutend. Dies ist anhand zweier Testdatensätze evaluiert worden, muss jedoch am jeweiligen Datenbestand im Einsatz überprüft werden. Eine vorsichtige Schätzung mag an dieser Stelle sein, dass für die Warenkörbe ein Datenumfang von zwei Jahren ausreichend ist, wobei eine Aufteilung in sechs bis zwölf Monate sinnvoll sein kann. Eine Überlappung sollte dabei eher gering gewählt werden. Für die Anzeige der Empfehlungen sollte eine Abwägung hinsichtlich der Differential Privacy erfolgen. Abhängig vom Nutzerverhalten und dem Regelwerk in der Bibliothek sind die Parameter Support und Konfidenz für die Warenkorbanalyse zu nutzen. Ein höherer Support geht mit einer höheren Anonymität einher. Für die Anforderung der Differential Privacy bietet das Konstrukt der randomisierten und eingeschränkten Empfehlungsrepräsentation einen geeigneten Rahmen.

Die drei Forschungsfragen konnten dahingehend beantwortet werden, dass das vorgestellte Vorgehen ein System nach dem Data-Privacy-By-Design-Konzept darstellt: von der Anonymisierung der Informationen für den Warenkorb über die Datentransformationen bis hin zur Ergebnispräsentation. Hierarchische Strukturen in den Daten lassen sich sehr gut nutzen und führen dann in einem anderen Kontext zu weiteren Empfehlungen. Auch das Vergessen von Informationen der Ausleihe führen nicht unbedingt zu schlechteren Ergebnissen. Insbesondere der Verlust von Zusammenhängen wirkt positiv auf die Empfehlungsmenge. Hier liegt die Begründung wahrscheinlich in der Tatsache, dass Nutzerpräferenzen sich über die Zeit ändern und die längeren Zusammenhänge eine eher untergeordnete Rolle spielen.

6.2 Ausblick

Einige Punkte müssen für zukünftige Untersuchungen offenbleiben. Zunächst ist das Empfehlungssystem hinsichtlich Nutzerakzeptanz und Effektivität zu überprüfen. Dabei sollten unterschiedliche Nutzergruppen berücksichtigt werden, denn sowohl Bibliothekare als auch Nutzer haben einen unterschiedlichen Fokus. Außerdem ist die Einbettung in den Discovery-Dienst eine zusätzliche Funktionalität, die diesen Dienst weiter anreichern würde. Auch an dieser Stelle kann eine Nutzerstudie durchgeführt werden, um weitere Optimierungspotentiale zu heben.

Eine Empfehlung, die nicht verfügbar ist, erzeugt eher Frustrationspotential. Daher sollte eine Anbindung an die Verfügbarkeitsschnittstelle ebenfalls in Erwägung gezogen werden. Dabei kann zugleich das hierarchische Konzept umgesetzt werden. So kann für den Fall, dass die neueste Auflage nicht verfügbar ist, auf eine andere verwiesen werden. Aufgrund der notwendigen Kommunikation über die Schnittstellen zwischen den unterschiedlichen Systemen, können neuere Ansätze der Kommunikationsstrategien an dieser Stelle ebenfalls untersucht werden.

Für die hierarchischen Beziehungen wurden manuelle Mapping-Dateien erstellt. Dies ist für ein Proof-of-Concept ausreichend, aber für eine Betriebslösung unzureichend. Nicht nur die kontinu-

ierlichen Neuzugänge, sondern auch die Änderungen am Datenbestand müssen gepflegt werden. Katalogdaten, wie die GND der DNB bieten eine gute Möglichkeit, die Daten weiter anzureichern. Aufgrund der Komplexität der Datenintegration und des Datenmodells muss geprüft werden, wie eine Anbindung erfolgen kann. Hierbei müssen Fragen der Datenqualität und der Anfrageverarbeitung betrachtet werden. Zudem muss das rudimentäre Datenbankschema der vorgestellten Lösung erweitert werden. Hier sind insbesondere die genutzten Identifizierer und die Repräsentationen für die Empfehlungsebene interessant. Zudem können hierarchische Methoden verwendet werden, die in der vorliegenden Arbeit nicht berücksichtigt wurden.

Offen bleibt auch die Frage, in welchen Abständen eine Aktualisierung des Empfehlungssystems erfolgen sollte. Hier sind Ansätze aus dem ETL denkbar. Auch der Einsatz von weiteren Pattern kann Verbesserungspotentiale hinsichtlich der Dokumentation der Prozesse, der Datenqualität und des Vorgehens erzielen. Dabei bleibt zu klären, inwieweit Aktualisierungszeitpunkte im betrieblichen Einsatz zu setzen sind. Ebenfalls offen ist die Thematik der inkrementellen im Vergleich zur vollständigen Aktualisierung.

Ein Datenhaltungskonzept, das sich für die gestellten Anforderungen ebenfalls anbietet, ist das Data Warehouse. In vielen Belangen, z.B. der DBS werden bereits Informationen genutzt, die eher dem Data Warehouse zuzuordnen sind. Deshalb bietet sich eine konzeptionelle Weiterentwicklung hin zu einem Datenlager auch für Data-Mining-Anwendungen an. Hier ist zu klären, welche Transformationen dadurch besser umgesetzt werden können. Auch Weiterentwicklungen, z.B. bei der Datenhaltung, dem Einsatz neuer Technologien oder neuen Datenstrukturen für multidimensionale Informationen bieten hinreichende Untersuchungsgegenstände.

Die Personalisierung des Empfehlungssystems ist in der vorliegenden Arbeit nicht thematisiert. Neben den Anforderungen an die Data Privacy müssen in diesem Kontext auch Anpassungen der Warenkorbanalyse erfolgen. Denn eine personalisierte Empfehlung setzt voraus, dass in den Warenkörben Attribute für die Personalisierung vorliegen. Dies erweitert den Lösungsraum, aber auch mehr-elementige Regeln können dann sinnvoll betrachtet werden. So können anhand der aktuellen Ausleihen dem Nutzer weitere Buchempfehlungen gegeben werden.

Letztlich ist es im Kontext der sozialen Medien ebenfalls möglich, Nutzerbewertungen unabhängig vom bibliothekarischen Kontext zu verwenden. So können Bewertungen genutzt werden, um die Empfehlungsmenge personalisierter oder umfangreicher zu gestalten. An dieser Stelle müssen unterschiedliche Empfehlungsstrategien zu einem hybriden Empfehlungssystem zusammengesetzt werden. Dies betrifft Änderungen hinsichtlich der Systemperformanz und Fragen der Datenhaltung und der Data Privacy.

Literaturverzeichnis

[1] Gesetz zum Schutz personenbezogener Daten der Bürger (Datenschutzgesetz Sachsen-Anhalt - DSG LSA) in der Fassung der Bekanntmachung vom 13. Januar 2016, Januar 2016.

[2] Verordnung (EU) 2016/679 des europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG (Datenschutz-Grundverordnung), April 2016.

[3] Bundesdatenschutzgesetz in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), das zuletzt durch Artikel 10 Absatz 2 des Gesetzes vom 31. Oktober 2017 (BGBl. I S. 3618) geändert worden ist, 2017.

[4] A. Aamodt und M. Nygård. Different roles and mutual dependencies of data, information, and knowledge –an AI perspective on their integration. *Data & Knowledge Engineering*, 16 (3):191–222, 1995.

[5] D. J. Abadi, P. A. Boncz, und S. Harizopoulos. Column-Oriented Database Systems. *Proceedings of the VLDB Endowment (PVLDB)*, 2(2):1664–1665, 2009.

[6] G. Adomavicius und A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, Juni 2005. ISSN 1041-4347. doi: 10.1109/TKDE.2005.99.

[7] R. Agrawal und R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, Seiten 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1-55860-153-8. URL www.vldb.org/conf/1994/P487.PDF.

[8] R. Agrawal und R. Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, Mai 2000. ISSN 0163-5808. doi: 10.1145/335191.335438. URL http://doi.acm.org/10.1145/335191.335438.

[9] R. Agrawal, S. P. Ghosh, T. Imielinski, B. R. Iyer, und A. N. Swami. An interval classifier for database mining applications. In *Proceedings of the 18th International Conference on Very Large Data Bases*, VLDB '92, Seiten 560–573, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. ISBN 1-55860-151-1. URL http://dl.acm.org/citation.cfm?id=645918.672486.

[10] R. Agrawal, T. Imielinski, und A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Seiten 207–216, Washington D.C., May 1993.

[11] F. Aldá und B. I. Rubinstein. The bernstein mechanism: Function release under differential privacy. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, Seiten 1705–1711. Association for the Advancement of Artificial Intelligence, 2017. URL https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14735.

[12] G. Antoshenkov. Dictionary-based order-preserving string compression. *The VLDB Journal*, 6(1):26–39, 1997. ISSN 1066-8888.

[13] K. Backhaus, B. Erichson, W. Plinke, und R. Weiber. *Multivariate Analysemethoden*. Springer, 11. auflage edition, 2011. ISBN 3-540-27870-2.

[14] S. Badsha, X. Yi, und I. Khalil. A practical privacy-preserving recommender system. *Data Science and Engineering*, 1(3):161–177, 2016.

[15] S. Badsha, X. Yi, I. Khalil, und E. Bertino. Privacy preserving user-based recommender system. In *Proc. IEEE 37th Int. Conf. Distributed Computing Systems (ICDCS)*, Seiten 1074–1083, Juni 2017. doi: 10.1109/ICDCS.2017.248.

[16] R. Ball. *Bibliometrie im Zeitalter von Open und Big Data: Das Ende des klassischen Indikatorenkanons*. Number 56 in b.i.t.online-Innovativ. Verlag Dinges & Frick, Wiesbaden, 2015. ISBN 978-3-934997-72-1.

[17] C. Basu, H. Hirsh, und W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, Seiten 714–720, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence. ISBN 0-262-51098-7. URL http://dl.acm.org/citation.cfm?id=295240.295795.

[18] C. Batini und M. Scannapieco. *Data Quality: Concepts, Methodologies and Techniques*. Data-Centric Systems and Applications. Springer, 2006. ISBN 978-3-540-33173-5.

[19] R. Bayer und E. McCreight. Organization and Maintenance of Large Ordered Indexes. *Acta Informatica*, 1(3):173–189, 1972.

[20] J. Beel und S. Langer. A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. In S. Kapidakis, C. Mazurek, und M. Werla, Hrsg., *Research and Advanced Technology for Digital Libraries*, Seiten 153–168, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24592-8. doi: 10.1007/978-3-319-24592-8_12.

[21] J. Beel, B. Gipp, S. Langer, M. Genzmehr, E. Wilde, A. Nürnberger, und J. Pitman. Introducing mr. dlib, a machine-readable digital library. In *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*, JCDL '11, Seiten 463–464, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0744-4. doi: 10.1145/1998076.1998187. URL http://doi.acm.org/10.1145/1998076.1998187.

[22] J. Beel, B. Gipp, S. Langer, und C. Breitinger. Paper recommender systems: A literature survey. *International Journal on Digital Libraries*, 17(4):305–338, 2016.

[23] B. Berendt und V. Köppen. *Improving Ranking by Respecting the Multidimensionality and Uncertainty of User Preferences*, Seiten 39–56. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-14000-6. doi: 10.1007/978-3-642-14000-6_3. URL https://doi.org/10.1007/978-3-642-14000-6_3.

[24] A. Beutel, E. H. Chi, Z. Cheng, H. Pham, und J. Anderson. Beyond globally optimal: Focused learning for improved recommendations. In *Proceedings of the 26th International Conference on World Wide Web*, Seiten 203–212. International World Wide Web Conferences Steering Committee, 2017.

[25] D. Billsus und M. J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2):147–180, Jun 2000. ISSN 1573-1391. doi: 10.1023/A:1026501525781. URL https://doi.org/10.1023/A:1026501525781.

[26] C. Binnig, S. Hildenbrand, und F. Färber. Dictionary-based order-preserving string compression for main memory column stores. In *SIGMOD*, Seiten 283–296. ACM, 2009.

[27] M. Bishop. *Computer security: art and science*. Addison-Wesley Professional, 2003. ISBN 0-201-44099-7.

[28] F. Böcker. Die Analyse des Kaufverbunds - Ein Ansatz zur bedarfsorientierten Warentypologie. *Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung (ZfbF)*, 27(5):290–306, 1975. ISSN 0341-2687.

[29] W. Böhm, A. Geyer-Schulz, M. Hahsler, und M. Jahn. Repeat-buying theory and its application for recommender services. In M. Schwaiger und O. Opitz, Hrsg., *Exploratory Data Analysis in Empirical Research*, Seiten 229–239, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-642-55721-7.

[30] T. Bollinger. Assoziationsregeln – Analyse eines Data Mining Verfahrens. *Informatik-Spektrum*, 19(5):257–261, Oct 1996. ISSN 1432-122X. doi: 10.1007/s002870050036. URL https://doi.org/10.1007/s002870050036.

[31] C. Borgelt. An implementation of the fp-growth algorithm. In *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, OSDM '05, Seiten 1–5, New York, NY, USA, 2005. ACM. ISBN 1-59593-210-0. doi: 10.1145/1133905.1133907. URL http://doi.acm.org/10.1145/1133905.1133907.

[32] C. Borgelt und R. Kruse. Induction of association rules: Apriori implementation. In W. Härdle und B. Rönz, Hrsg., *Proceedings in Computational Statistics*, Seiten 395–400. Springer, 2002. doi: 10.1007/978-3-642-57489-4_59. URL http://dx.doi.org/10.1007/978-3-642-57489-4_59.

[33] Y. Boztuğ und N. Silberhorn. Modellierungsansätze in der Warenkorbanalyse im Überblick. *Journal für Betriebswirtschaft*, 56(2):105–128, Jun 2006. ISSN 1614-631X. doi: 10.1007/s11301-006-0008-5. URL https://doi.org/10.1007/s11301-006-0008-5.

[34] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, und F. Yergeau. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, November 2008.

[35] S. Brin, R. Motwani, J. D. Ullman, und S. Tsur. Dynamic itemset counting and implication rules for market basket data. *SIGMOD Rec.*, 26(2):255–264, Juni 1997. ISSN 0163-5808. doi: 10.1145/253262.253325. URL http://doi.acm.org/10.1145/253262.253325.

[36] D. Broneske, V. Köppen, G. Saake, und M. Schäler. Accelerating multi-column selection predicates in main-memory - the elf approach. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, Seiten 647–658, April 2017. doi: 10.1109/ICDE.2017.118.

[37] BSI. Kryptographische Verfahren: Empfehlungen und Schlüssellangen. Technical Report TR-02102-1, Bundesamt für Sicherheit in der Informationstechnik, Februar 2017.

[38] P. Buono und M. F. Costabile. Visualizing association rules in a framework for visual data mining. In M. Hemmje, C. Niederée, und T. Risse, Hrsg., *From Integrated Publication and Information Systems to Information and Knowledge Environments: Essays Dedicated to Erich J. Neuhold on the Occasion of His 65th Birthday*, Seiten 221–231, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31842-2. doi: 10.1007/978-3-540-31842-2_22. URL https://doi.org/10.1007/978-3-540-31842-2_22.

[39] R. Burke. The wasabi personal shopper: A case-based recommender system. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, Seiten 844–849, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence. ISBN 0-262-51106-1. URL http://www.aaai.org/Library/IAAI/1999/iaai99-119.php.

[40] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, Nov 2002. ISSN 1573-1391. doi: 10.1023/A:1021240730564. URL https://doi.org/10.1023/A:1021240730564.

[41] R. Burke. Hybrid web recommender systems. In *The adaptive Web*, Seiten 377–408. Springer, Berlin Heidelberg, 2007.

[42] R. K. Chellappa und R. G. Sin. Personalization versus privacy: An empirical examination of the online consumer's dilemma. *Information Technology and Management*, 6(2):181–202, Apr 2005. ISSN 1573-7667. doi: 10.1007/s10799-005-5879-y. URL https://doi.org/10.1007/s10799-005-5879-y.

[43] L. Chen und Q. Mei. Mining frequent items in data stream using time fading model. *Information Sciences*, 257:54–69, 2014. ISSN 0020-0255. doi: 10.1016/j.ins.2013.09.007. URL http://www.sciencedirect.com/science/article/pii/S0020025513006403.

[44] N. Choi und S. Joo. Booklovers' world: An examination of factors affecting continued usage of social cataloging sites. *Journal of the Association for Information Science and Technology*, 67(12):3022–3035, 2015. doi: 10.1002/asi.23556.

[45] P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Springer, 2012. ISBN 978-3-642-31164-2.

[46] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, und M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *ACM SIGIR '99 Workshop*

on Recommender Systems: Algorithms and Evaluation, 1999. URL https://www.csee.umbc.edu/~ian/sigir99-rec/.

[47] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6): 377–387, Juni 1970. ISSN 0001-0782. doi: 10.1145/362384.362685. URL http://doi.acm.org/10.1145/362384.362685.

[48] E. F. Codd, S. B. Codd, und C. T. Salley. Providing OLAP (Online Analytical Processing) to User-Analysts: An IT Mandate. E.F. Codd Associates, 1993.

[49] E. Cragg und K. Birkwood. Beyond books: what it takes to be a 21st century librarian. *Guardian*, 2011. URL https://www.theguardian.com/careers/job-of-21st-century-librarian. letzter Zugriff: 19.02.2018.

[50] T. Dalenius. Finding a needle in a haystack or identifying anonymous census records. *Journal of Official Statistics*, 2(3):329–336, 1986. ISSN 0282-423X.

[51] R. Decker. Empirischer Vergleich alternativer Ansätze zur Verbundanalyse im Marketing. In E. Schumacher und K. Streichfuss, Hrsg., *Proceedings der 5. Konferenz der SAS-Anwender in Forschung und Entwicklung (KSFE)*, Seiten 99–110. Universität Hohenheim, 2001.

[52] R. Decker und K. Monien. Market basket analysis with neural gas networks and self-organising maps. *Journal of Targeting, Measurement and Analysis for Marketing*, 11(4):373–386, May 2003. ISSN 1479-1862. doi: 10.1057/palgrave.jt.5740092. URL https://doi.org/10.1057/palgrave.jt.5740092.

[53] W. H. DeLone und E. R. McLean. Information systems success: The quest for the dependent variable. *Information Systems Research*, 3(1):60–95, 1992. doi: 10.1287/isre.3.1.60.

[54] B. den Boer und A. Bosselaers. Collisions for the compression function of MD5. In *Workshop on the Theory and Application of of Cryptographic Techniques*, Seiten 293–304. Springer, 1993.

[55] J. Dittmann. *Digitale Wasserzeichen: Grundlagen, Verfahren, Anwendungsgebiete*. Xpert.press. Springer-Verlag, 2000. ISBN 3-540-66661-3.

[56] B. Drees. Text und Data Mining: Herausforderungen und Möglichkeiten für Bibliotheken. *Perspektive Bibliothek*, 5(1):49–73, 2016.

[57] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, und I. Wegener, Hrsg., *Automata, Languages and Programming*, Seiten 1–12, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-35908-1.

[58] C. Dwork, F. McSherry, K. Nissim, und A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi und T. Rabin, Hrsg., *Theory of Cryptography*, Seiten 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5. doi: 10.1007/11681878_14.

[59] D. Eastlake und T. Hansen. US secure hash algorithms (SHA and SHA-based HMAC and HKDF). Technical Report RFC 6234, Internet Engineering Task Force, Mai 2011.

[60] A. S. C. Ehrenberg. *Repeat-buying : facts, theory and applications*. Charles Griffin & Company, new ed edition, 1988. ISBN 0-85264-287-3.

[61] G. Engelmann. Discovery-Systeme: Erfahrungen im Umgang mit Lukida. *VZG aktuell*, 2016 (2):21–25, 2016.

[62] G. Ertek und A. Demiriz. A framework for visualizing association mining results. In A. Levi, E. Savaş, H. Yenigün, S. Balcısoy, und Y. Saygın, Hrsg., *Computer and Information Sciences – ISCIS 2006*, Seiten 593–602, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-47243-8.

[63] D. Fasel und A. Meier, Hrsg. *Big Data*. Edition HMD. Springer Vieweg, Wiesbaden, 2016. ISBN 9783658115883. Literaturangaben.

[64] U. Fayyad, G. Piatetsky-Shapiro, und P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54, 1996. ISSN 0738-4602. URL https://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/1230/1131.

[65] S. Feyer, S. Siebert, B. Gipp, A. Aizawa, und J. Beel. Integration of the scientific recommender system mr. dlib into the reference manager jabref. In *European Conference on Information Retrieval*, Seiten 770–774. Springer, 2017.

[66] S. Fu, Y. Zhang, und S. Minn. On the recommender system for university library. In *International Association for Development of the Information Society*, Seiten 215–222. ERIC, 2013. ISBN 978-972-8939-88-5.

[67] K. Fuchs-Kittowski, H. Parthey, W. Umstätter, und R. Wagner-Döbler, Hrsg. *Organisationsinformatik und digitale Bibliothek in der Wissenschaft*. GeWiF, Berlin, 2. auflage edition, 2010. ISBN 978-3-934682-53-5.

[68] K. Gärtner. *Innovationspreis 2012-Analyse von Recommendersystemen in Deutschland: Literaturstudie*, volume 38. BIT Verlag, 2012.

[69] A. Geyer-Schulz, M. Hahsler, und M. Jahn. Recommendations for virtual universities from observed user behavior. In W. Gaul und G. Ritter, Hrsg., *Classification, Automation, and New Media*, Seiten 273–280, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-642-55991-4.

[70] D. Goldberg, D. Nichols, B. M. Oki, und D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, Dezember 1992. ISSN 0001-0782. doi: 10.1145/138859.138867. URL http://doi.acm.org/10.1145/138859.138867.

[71] J. Gray. The transaction concept: Virtues and limitations (invited paper). In *Proceedings of the Seventh International Conference on Very Large Data Bases - Volume 7*, VLDB '81, Seiten 144–154. VLDB Endowment, 1981. URL http://dl.acm.org/citation.cfm?id=1286831.1286846.

[72] G. K. Gupta, A. Strehl, und J. Ghosh. Distance based clustering of association rules. In *Intelligent Engineering Systems Through Artificial Neural Networks (Proceedings of ANNIE 1999)*, Seiten 759–764. ASME Press, 1999.

[73] A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, Seiten 47–57, Boston, NJ, 1984.

[74] M. Hahsler und R. Karpienko. Visualizing association rules in hierarchical groups. *Journal of Business Economics*, 87(3):317–335, Apr 2017. ISSN 1861-8928. doi: 10.1007/s11573-016-0822-8. URL https://doi.org/10.1007/s11573-016-0822-8.

[75] M. Hahsler, B. Grün, und K. Hornik. A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, 2005. URL https://www.jstatsoft.org/article/view/v014i15.

[76] M. Hahsler, S. Chelluboina, K. Hornik, und C. Buchta. The arules r-package ecosystem: Analyzing interesting patterns from large transaction datasets. *Journal of Machine Learning Research*, 12:2021–2025, 2011. URL http://jmlr.csail.mit.edu/papers/v12/hahsler11a.html.

[77] M. Hahsler, C. Buchta, B. Gruen, und K. Hornik. *arules: Mining Association Rules and Frequent Itemsets*, 2018. URL https://CRAN.R-project.org/package=arules. R package version 1.6-0.

[78] J. Han und M. Kamber. *Data Mining – Concepts and Techniques*. Morgan Kaufmann Publishers, 2 edition, 2006.

[79] J. Han, J. Pei, und Y. Yin. Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29(2):1–12, 2000. ISSN 0163-5808. doi: 10.1145/335191.335372.

[80] M. E. Henderson. *Data management : a practical guide for librarians*. Number 28 in Practical guides for librarians. Rowman & Littlefield, London, 2017. ISBN 9781442264380 | 978-1-4422-6438-0 | 9781442264397 (electronic) | 978-1-4422-6439-7. xvii, 195 Seiten, Diagramme, 28 cm.

[81] W. Hill, L. Stead, M. Rosenstein, und G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, Seiten 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-84705-1. doi: 10.1145/223904.223929. URL http://dx.doi.org/10.1145/223904.223929.

[82] P. Hitzler, M. Krötzsch, und S. R. andYork Sure. *Semantic Web: Grundlagen*. examen.press. Springer, 2008. ISBN 978-3-540-33993-9. doi: 10.1007/978-3-540-33994-6.

[83] H. Hruschka. Bestimmung der Kaufverbundenheit mit Hilfe eines probabilistischen Meßmodells. *Schmalenbachs Zeitschrift für die betriebswirtschaftliche Forschung (ZfbF)*, 43(5):418–434, 1991.

[84] W. H. Inmon. *Building the Data Warehouse*. John John Wiley & Sons, 3 edition, 2002. ISBN 0-471-08130-2.

[85] ISO99a. *ISO/IEC International Standard (IS) Database Language SQL – Part 1: SQL/Framework, ISO/IEC 9075-1:2008 (E)*, September 2008.

[86] D. Jannach, M. Zanker, A. Felfernig, und G. Friedrich. *Recommender systems*. Safari Tech Books Online. Cambridge Univ. Press, Cambridge [u.a.], 2011. ISBN 9780511911309. URL http://proquest.safaribooksonline.com/9780521493369. Parallel als Druckausg. erschienen.

[87] A. J. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. L. Lagendijk, und Q. Tang. Privacy in recommender systems. In *Social media retrieval*, Seiten 263–281. Springer, 2013.

[88] A. Jennings und H. Higuchi. A user model neural network for a personal news service. *User Modeling and User-Adapted Interaction*, 3(1):1–25, Mar 1993. ISSN 1573-1391. doi: 10.1007/BF01099423. URL https://doi.org/10.1007/BF01099423.

[89] K. Kemner-Heek. Konzeption und Angebot zukünftiger Bibliotheksmanagementsysteme: Bestandsaufnahme und Analyse. Master's thesis, Fachhochschule Köln, 2011.

[90] M. A. Kerr und S. E. Symons. Computerized presentation of text: Effects on children's reading of informational material. *Reading and Writing*, 19(1):1–19, Feb 2006. ISSN 1573-0905. doi: 10.1007/s11145-003-8128-y. URL https://doi.org/10.1007/s11145-003-8128-y.

[91] B. N. Keshavamurthy, A. M. Khan, und D. Toshniwal. Privacy preserving association rule mining over distributed databases using genetic algorithm. *Neural Computing and Applications*, 22(1):351, Mai 2013. ISSN 1433-3058. doi: 10.1007/s00521-013-1343-9. URL http://dx.doi.org/10.1007/s00521-013-1343-9.

[92] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, und A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proceedings of the Third International Conference on Information and Knowledge Management*, CIKM '94, Seiten 401–407, New York, NY, USA, 1994. ACM. ISBN 0-89791-674-3. doi: 10.1145/191246.191314. URL http://doi.acm.org/10.1145/191246.191314.

[93] B. P. Knijnenburg und A. Kobsa. Making decisions about privacy: information disclosure in context-aware recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(3):20, 2013.

[94] B. P. Knijnenburg, A. Kobsa, und H. Jin. Dimensionality of information disclosure behavior. *International Journal of Human-Computer Studies*, 71(12):1144–1162, 2013.

[95] Y. Kodratoff. *Comparing Machine Learning and Knowledge Discovery in DataBases: An Application to Knowledge Discovery in Texts*, Seiten 1–21. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-44673-6. doi: 10.1007/3-540-44673-7_1.

[96] V. Köppen, B. Brüggemann, und B. Berendt. Designing Data Integration: The ETL Pattern Approach. *Cepis Upgrade*, 13(3):49–55, Juli 2011.

[97] V. Köppen, G. Saake, und K.-U. Sattler. *Data Warehouse Technologien*. mitp, 2. auflage edition, 2014.

[98] M. Kracht. *Wissen und materiale Kultur*. Barton academics. Barton Verlag, Weilerswist, 2017. ISBN 9783934648142. Literaturverzeichnis: Seite 186-192.

[99] B. Krulwich. Lifestyle Finder: Intelligent user profiling using large-scale demographic data. *AI MAGAZINE*, 18(2):37–45, 1997.

[100] J. Lee und C. Clifton. How much is enough? choosing ϵ for differential privacy. In X. Lai, J. Zhou, und H. Li, Hrsg., *Information Security*, Seiten 325–340, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-24861-0. doi: 10.1007/978-3-642-24861-0_22.

[101] N. Li, T. Li, und S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, Seiten 106–115, April 2007. doi: 10.1109/ICDE.2007.367856.

[102] S. Loudcher, W. Jakawat, E. P. S. Morales, und C. Favre. Combining olap and information networks for bibliographic data analysis: a survey. *Scientometrics*, 103(2):471–487, May 2015. ISSN 1588-2861. doi: 10.1007/s11192-015-1539-0. URL https://doi.org/10.1007/s11192-015-1539-0.

[103] Z. Lu und H. Shen. An accuracy-assured privacy-preserving recommender system for internet commerce. *Computer Science and Information Systems*, 12(4):1307–1326, 2015. doi: https://doi.org/10.2298/CSIS140725056L.

[104] A. Machanavajjhala, J. Gehrke, D. Kifer, und M. Venkitasubramaniam. ℓ-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, Seite 24, 2006. doi: 10.1109/ICDE.2006.1. URL http://doi.ieeecomputersociety.org/10.1109/ICDE.2006.1.

[105] M. Malirsch. *Big Data: Büchse der Pandora*. Re Di Roma-Verlag, 2013. ISBN 978-3-86870-598-0.

[106] A. Mangen, B. R. Walgermo, und K. Brønnick. Reading linear texts on paper versus computer screen: Effects on reading comprehension. *International Journal of Educational Research*, 58:61 – 68, 2013. ISSN 0883-0355. doi: https://doi.org/10.1016/j.ijer.2012.12.002. URL http://www.sciencedirect.com/science/article/pii/S0883035512001127.

[107] F. Manola, E. Miller, und B. McBride. RDF primer 1.1. W3C Recommendation, 2014. URL https://www.w3.org/TR/rdf11-primer/.

[108] P. Matuszyk und M. Spiliopoulou. Selective forgetting for incremental matrix factorization in recommender systems. In S. Džeroski, P. Panov, D. Kocev, und L. Todorovski, Hrsg., *Discovery Science*, Seiten 204–215, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11812-3.

[109] P. Matuszyk, J. Vinagre, M. Spiliopoulou, A. M. Jorge, und J. Gama. Forgetting techniques for stream-based matrix factorization in recommender systems. *Knowledge and Information Systems*, 55(2):275–304, May 2018. ISSN 0219-3116. doi: 10.1007/s10115-017-1091-8. URL https://doi.org/10.1007/s10115-017-1091-8.

[110] D. W. McMillan und D. M. Chavis. Sense of community: A definition and theory. *Journal of Community Psychology*, 14(1):6–23, 1986. doi: 10.1002/1520-6629(198601)14:1<6::AID-JCOP2290140103>3.0.CO;2-I. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/1520-6629%28198601%2914%3A1%3C6%3A%3AAID-JCOP2290140103%3E3.0.CO%3B2-I.

[111] D. McSherry. Diversity-conscious retrieval. In S. Craw und A. Preece, Hrsg., *Advances in Case-Based Reasoning*, Seiten 219–233, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-46119-7.

[112] P. Melville, R. J. Mooney, und R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*,

Seiten 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence. ISBN 0-262-51129-0. URL https://www.cs.utexas.edu/~ml/papers/cbcf-aaai-02.pdf.

[113] L. H. Mendes, J. Quiñonez-Skinner, und D. Skaggs. Subjecting the catalog to tagging. *Library Hi Tech*, 27(1):30–41, 2009. doi: 10.1108/07378830910942892.

[114] R. Mendes und J. P. Vilela. Privacy-preserving data mining: Methods, metrics, and applications. *IEEE Access*, 5:10562–10582, 2017. doi: 10.1109/ACCESS.2017.2706947.

[115] I. Mohallick. User privacy in recommender systems. Master's thesis, NTNU, 2017.

[116] M. Mönnich und M. Spiering. Einsatz von BibTip als Recommendersystem im Bibliothekskatalog. *Bibliotheksdienst*, 42(1):54–59, 2008.

[117] M. W. Mönnich und M. Spiering. Bibtip: Recommendersystem für den Bibliothekskatalog. *EUCOR-Bibliotheksinformationen*, 30:4–8, 2007.

[118] R. M. Müller und H.-J. Lenz. *Business Intelligence*. examen.press. Springer Springer Vieweg, 2013. ISBN 978-3-642-35559-2. doi: 10.1007/978-3-642-35560-8.

[119] M. S. Mythili und A. R. M. Shanavas. Article: Performance evaluation of Apriori and FP-Growth algorithms. *International Journal of Computer Applications*, 79(10):34–37, October 2013. ISSN 0975-8887. doi: 10.5120/13779-1650.

[120] O. Netzer, R. Feldman, J. Goldenberg, und M. Fresko. Mine your own business: Market-structure surveillance through text mining. *Marketing Science*, 31(3):521–543, 2012. ISSN 0732-239. doi: 10.1287/mksc.1120.0713. URL https://doi.org/10.1287/mksc.1120.0713.

[121] A. W. Neumann. *Recommender Systems for Information Providers: Designing Customer Centric Paths to Information*. Contributions to Management Science. Springer, 2009. ISBN 978-3-7908-2134-5.

[122] E. R. Omiecinski. Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):57–69, Jan 2003. ISSN 1041-4347. doi: 10.1109/TKDE.2003.1161582.

[123] J. S. Park, M.-S. Chen, und P. S. Yu. An effective hash-based algorithm for mining association rules. *SIGMOD Rec.*, 24(2):175–186, Mai 1995. ISSN 0163-5808. doi: 10.1145/568271.223813. URL http://doi.acm.org/10.1145/568271.223813.

[124] Z. Pawlak. Rough sets. *International Journal of Computer & Information Sciences*, 11(5): 341–356, Oct 1982. ISSN 1573-7640. doi: 10.1007/BF01001956.

[125] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5):393–408, Dec 1999. ISSN 1573-7462. doi: 10.1023/A:1006544522159. URL https://doi.org/10.1023/A:1006544522159.

[126] H. Plattner und A. Zeier. *In-Memory Data Management – Technology and Applications*. Springer-Verlag, 2 edition, 2012.

[127] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL https://www.R-project.org/.

[128] A. Richards und B. Sen. An investigation into the viability of librarything for promotional and user engagement purposes in libraries. *Library Hi Tech*, 31(3):493–519, 2013. ISSN 0737-8831. doi: 10.1108/lht-03-2013-0034.

[129] J. Riedl. Personalization and privacy. *IEEE Internet Computing*, 5(6):29–31, Nov 2001. ISSN 1089-7801. doi: 10.1109/4236.968828.

[130] R. L. Rivest. RFC 1321: The MD5 message-digest algorithm. Technical report, MIT Laboratory for Computer Science and RSA Data Security, Inc., 1992.

[131] G. J. Russell, S. Ratneshwar, A. D. Shocker, D. Bell, A. Bodapati, A. Degeratu, L. Hildebrandt, N. Kim, S. Ramaswami, und V. H. Shankar. Multiple-category decision-making: Review and synthesis. *Marketing Letters*, 10(3):319–332, Aug 1999. ISSN 1573-059X. doi: 10.1023/A:1008143526174. URL https://doi.org/10.1023/A:1008143526174.

[132] G. Saake, I. Schmitt, und C. Türker. *Objektdatenbanken — Konzepte, Sprachen, Architekturen*. International Thomson Publishing, Bonn, 1997. ISBN 3-8266-0258-7.

[133] G. Saake, K.-U. Sattler, und A. Heuer. *Datenbanken: Konzepte und Sprachen*. HJR-Verlag, Heidelberg, 5 edition, 2013. ISBN 978-3826694530.

[134] J. B. Schafer, D. Frankowski, J. Herlocker, und S. Sen. Collaborative filtering recommender systems. In *The Adaptive Web*. Springer, Januar 2007. ISBN 978-3-540-72078-2. doi: 10.1007/978-3-540-72079-9_9. URL http://dx.doi.org/10.1007/978-3-540-72079-9_9.

[135] A. Schuster, R. Wolff, und B. Gilburd. Privacy-preserving association rule mining in large-scale distributed systems. In *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*, Seiten 411–418. IEEE, 2004.

[136] P. B. Seetharaman, S. Chib, A. Ainslie, P. Boatwright, T. Chan, S. Gupta, N. Mehta, V. Rao, und A. Strijnev. Models of multi-category choice behavior. *Marketing Letters*, 16(3):239–254, Dec 2005. ISSN 1573-059X. doi: 10.1007/s11002-005-5888-y. URL https://doi.org/10.1007/s11002-005-5888-y.

[137] X. Shang, K.-U. Sattler, und I. Geist. SQL based frequent pattern mining with FP-growth. In *Applications of Declarative Programming and Knowledge Management*, Seiten 32–46. Springer, 2005.

[138] L. M. Singer und P. A. Alexander. Reading across mediums: Effects of reading digital and print texts on comprehension and calibration. *The Journal of Experimental Education*, 85(1):155–172, 2017. ISSN 0022-0973. doi: 10.1080/00220973.2016.1143794.

[139] Y. Sismanis, A. Deligiannakis, N. Roussopoulos, und Y. Kotidis. Dwarf: Shrinking the petacube. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, Seiten 464–475, New York, NY, USA, 2002. ACM. ISBN 1-58113-497-5. doi: 10.1145/564691.564745. URL http://doi.acm.org/10.1145/564691.564745.

[140] B. Smyth und P. Cotter. A personalised TV listings service for the digital TV age. *Knowledge-Based Systems*, 13(2):53 – 59, 2000. ISSN 0950-7051. doi: https://doi.org/10.1016/S0950-7051(00)00046-0. URL http://www.sciencedirect.com/science/article/pii/S0950705100000460.

[141] R. Srikant und R. Agrawal. Mining quantitative association rules in large relational tables. In *SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, Seiten 1–12. ACM, 1996. ISBN 0-89791-794-4. doi: 10.1145/233269.233311.

[142] R. Srikant und R. Agrawal. Mining generalized association rules. *Future generation computer systems*, 13(2-3):161–180, 1997. ISSN 0167-739X. doi: 10.1016/S0167-739X(97)00019-8.

[143] V. Srivastava. Multi-agent modeling of risk-aware and privacy-preserving recommender systems. Master's thesis, University of Waterloo, 2017.

[144] L. Sweeney. Simple demographics often identify people uniquely. Technical Report 3, Carnegie Mellon University, Pittsburgh, 2000.

[145] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[146] P.-N. Tan, V. Kumar, und J. Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293 – 313, 2004. ISSN 0306-4379. doi: https://doi.org/10.1016/S0306-4379(03)00072-3. URL http://www.sciencedirect.com/science/article/pii/S0306437903000723. Knowledge Discovery and Data Mining (KDD 2002).

[147] C. Tex, M. Schäler, und K. Böhm. Towards meaningful distance-preserving encryption. In *SSDBMÍ18: 30th International Conference on Scientific and Statistical Database Management*, Bozen-Bolzano, Italy, Juli 2018. ACM. doi: 10.1145/3221269.3223029.

[148] H. Toivonen. Sampling large databases for association rules. In *VLDB*, Seiten 134–145, Mumbai, India, 1996.

[149] A. Vellino. A comparison between usage-based and citation-based methods for recommending scholarly research articles. *Proceedings of the Association for Information Science and Technology*, 47(1):1–2, 2010. ISSN 0044-7870. doi: 10.1002/meet.14504701330.

[150] A. Vellino. Recommending research articles using citation data. *Library Hi Tech*, 33(4): 597–609, 2015. ISSN 0737-8831. doi: 10.1108/lht-06-2015-0063.

[151] E. Volokh. Personalization and privacy. *Commun. ACM*, 43(8):84–88, August 2000. ISSN 0001-0782. doi: 10.1145/345124.345155. URL http://doi.acm.org/10.1145/345124.345155.

[152] S. Wakeling, P. Clough, B. Sen, und L. S. Connaway. "readers who borrowed this also borrowed. . ."?: recommender systems in uk libraries. *Library Hi Tech*, 30(1):134–150, 2012. ISSN 0737-8831. doi: 10.1108/07378831211213265.

[153] L. Wenige und J. Ruhland. Retrieval by recommendation: using LOD technologies to improve digital library search. *International Journal on Digital Libraries*, 2017. ISSN 1432-5012. doi: 10.1007/s00799-017-0224-8.

[154] H. Wickham. The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, 40(1):1–29, 2011. URL http://www.jstatsoft.org/v40/i01/.

[155] J. Xun, L.-c. Xu, und L. Qi. Association rules mining algorithm based on rough set. In *Information Technology in Medicine and Education (ITME), 2012 International Symposium on*, volume 1, Seiten 361–364. IEEE, 2012.

[156] M. J. Zaki, S. Parthasarathy, M. Ogihara, und W. Li. New algorithms for fast discovery of association rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, KDD'97, Seiten 283–286. AAAI Press, 1997. URL http://dl.acm.org/citation.cfm?id=3001392.3001454.

[157] B. Zhang, N. Wang, und H. Jin. Privacy concerns in online recommender systems: influences of control and user data input. In *Symposium on Usable Privacy and Security (SOUPS)*, Seiten 159–173, 2014.

[158] C. Zhang und S. Zhang. *Association rule mining: models and algorithms*. Number 2307 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 2002. ISBN 3-540-43533-6.

[159] S. Zhang, L. Chen, und L. Tu. Frequent items mining on data stream based on time fading factor. In *2009 International Conference on Artificial Intelligence and Computational Intelligence*, Seiten 336–340. IEEE, 2009. doi: 10.1109/AICI.2009.369.

[160] Z. Zhang. Feeling the sense of community in social networking usage. *IEEE Transactions on Engineering Management*, 57(2):225–239, May 2010. ISSN 0018-9391. doi: 10.1109/TEM.2009.2023455.

[161] T. Zhu, G. Li, W. Zhou, und P. S. Yu. *Preliminary of Differential Privacy*, Seiten 7–16. Springer International Publishing, Cham, 2017. ISBN 978-3-319-62004-6. doi: 10.1007/978-3-319-62004-6_2. URL https://doi.org/10.1007/978-3-319-62004-6_2.

Abkürzungsverzeichnis

ANSI American National Standards Institute 16

ASCII American Standard Code for Information Interchange 15

BSI Bundesamt für Sicherheit in der Informationstechnik 54

CSV Comma-Separated-Values 15

DBMS Datenbankmanagementsystem 17

DNB Deutsche Nationalbibliothek 64

DBS Deutsche Bibliotheksstatistik 53

EPN Exemplar Produktionsnummer 6

ETL Extract-Transform-Load 36

FP-Baum Frequent-Pattern-Baum 45

GBV Gemeinsamer Bibliotheksverbund 35

GND Gemeinsame Normdatei 64

GVK Gemeinsamer Virtueller Katalog 10

JSON JavaScript Object Notation 16

KDP Knowledge Discovery Prozess 25

LBS Lokales Bibliothekssystem 7

LHS Left-hand Side 58

LoC Library of Congress 30

LOD Linked Open Data 31

OLAP Online Analytical Processing 18

OLTP Online Transaction Processing 18

OPAC Online Public Access Catalog 10